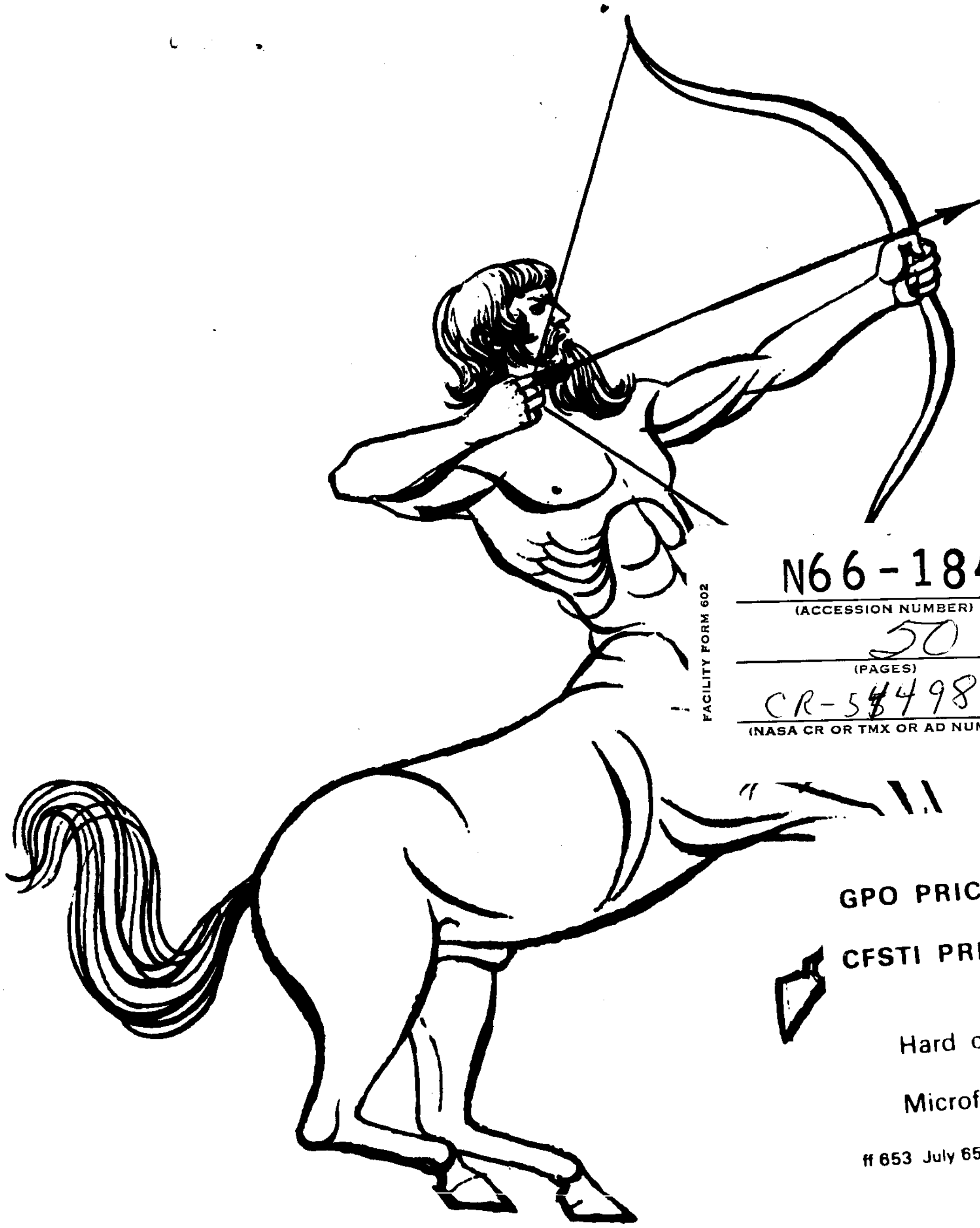
CR-54498

McLaughlin apd

GDA-BTD64-016
20 JANUARY 1964





# CENTAUR GUIDANCE ANALYSIS GENERAL TECHNIQUES

GENERAL DYNAMICS | ASTRONAUTICS

CENTAUR GUIDANCE ANALYSIS

GENERAL TECHNIQUES

GDA-BTD64-016

20 January 1964

CONTRACT NAS3-3232

Prepared by D. F. Flowers
Senior Research Engineer

Checked by H. Kienberger
Research Engineer

Prepared by R. Day
Research Group Engineer
Centaur Guidance Analysis

Approved by S. P. Hartzberg
Asst. Chief Engineer
Centaur Guidance System



GD
GENERAL DYNAMICS | ASTRONAUTICS

## FOREWORD

General techniques used to develop guidance equations for Centaur missions are presented herein as an aid to their understanding. Included are presentations of techniques used in developing guidance criteria, developing detailed guidance equations, and mechanizing the equations to be applicable to the guidance hardware being used. To provide a more complete presentation, general discussions of hardware error analysis techniques performed prior to flight and of guidance evaluation following flight tests are also included.

## TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

## SECTION 1

## GENERAL DESCRIPTION OF GUIDANCE SYSTEM

The guidance system is required to provide the guidance necessary to achieve all mission requirements. This includes, but is not restricted to the following:

a. Provide discretes to terminate powered flight phases meeting the required cutoff conditions.

b. Provide steering signals to guide all powered phases of flight following an open-loop booster section.

c. Provide trajectory data for telemetering.

d. Provide sequencing discretes for starting reorientation after coast, and for initiating spacecraft separation after injection.

To satisfy these requirements, the vehicleborne guidance system has the following capabilities:

a. A reference frame stabilized in inertial space from which three mutually perpendicular accelerations can be measured.

b. A means of integrating the accelerations measured in the inertial reference frame to obtain vehicle velocity and position in inertial space.

c. A means of determining the steering signals required to guide the vehicle to the proper kinematic conditions at the end of each phase of powered flight.

d. A means of transforming the steering signals from the inertial reference frame to the instantaneous vehicle coordinate system employed by the autopilot.

e. A means of determining when discretes should be issued.

f. A means of generating the discretes when required.

g. A means of providing trajectory data in a form suitable to telemetry system requirements.

The equipment needed to meet these requirements consists of an inertial platform, navigation computer, associated electronics, and power supplies. A functional block diagram of the inertial guidance system, including key elements of the autopilot is shown in Figure 1.

The Centaur inertial platform is a four-gimbal, all-attitude stable platform with three single-degree-of-freedom rate integrating gyros and three pendulous pulse-rebalanced

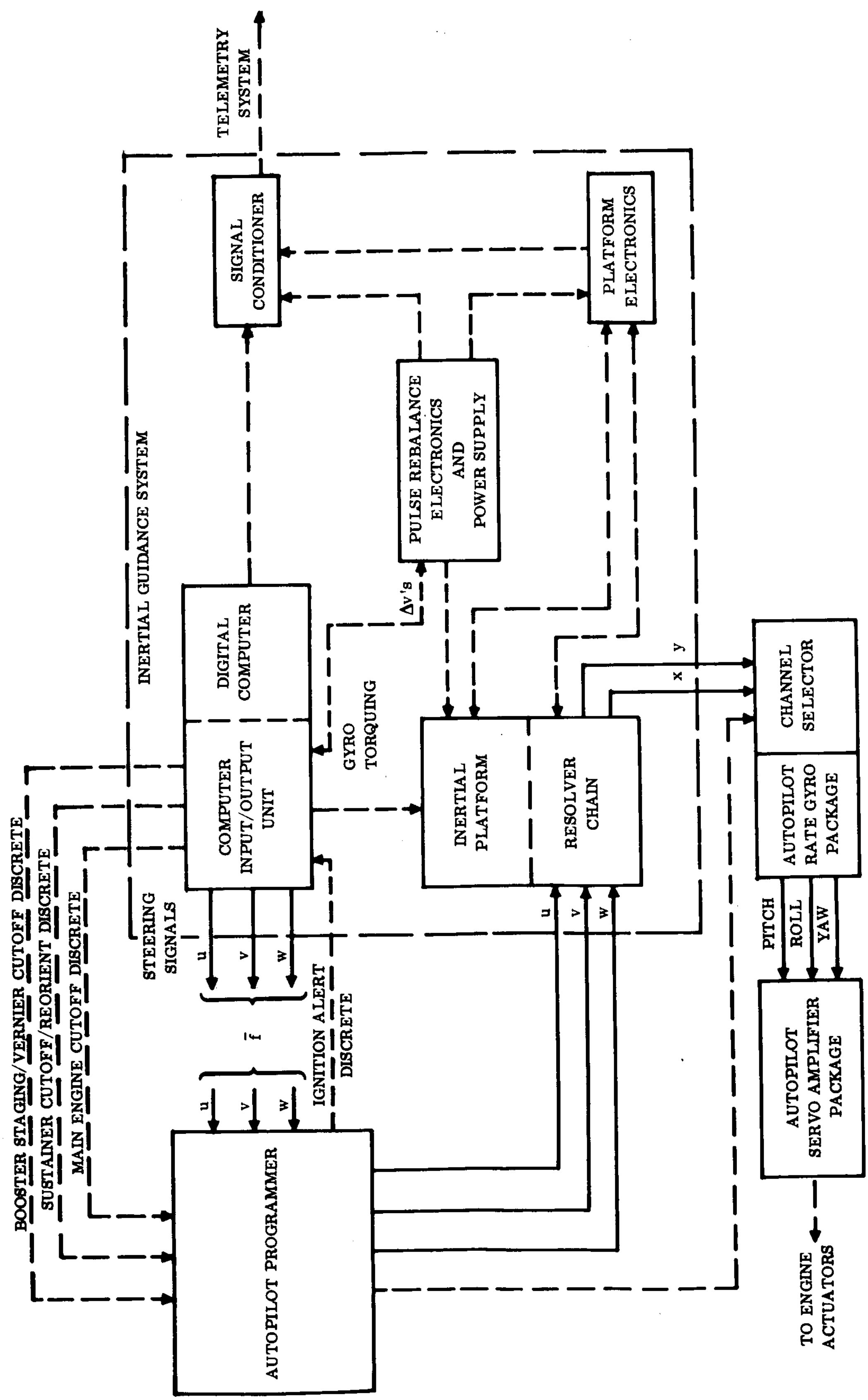



Figure 1. Functional Block Diagram of Inertial Guidance System

accelerometers mounted on the inner gimbal (see Figure 2). The gimbals are servo-driven by the outputs of the gyros to maintain the inner gimbal fixed in inertial space. The accelerometers provide outputs proportional to the accelerations measured along the three mutually perpendicular axes of the inertial reference frame maintained by the gyros and the servo-driven gimbals.

Mounted on the four gimbals are resolvers which provide a voltage vector coordinate transformation from the inertial axes to the vehicle axes. The resolver chain is used to transform the guidance steering vector from one coordinate system to the other.

The pulse rebalance electronics provide pulses for nulling the accelerometer pendulum angle under the influence of an external thrust acceleration. It also provides pulses to the navigation computer, the weight of each pulse being equal to a nominal value of 0.1 ft/sec increment in velocity.

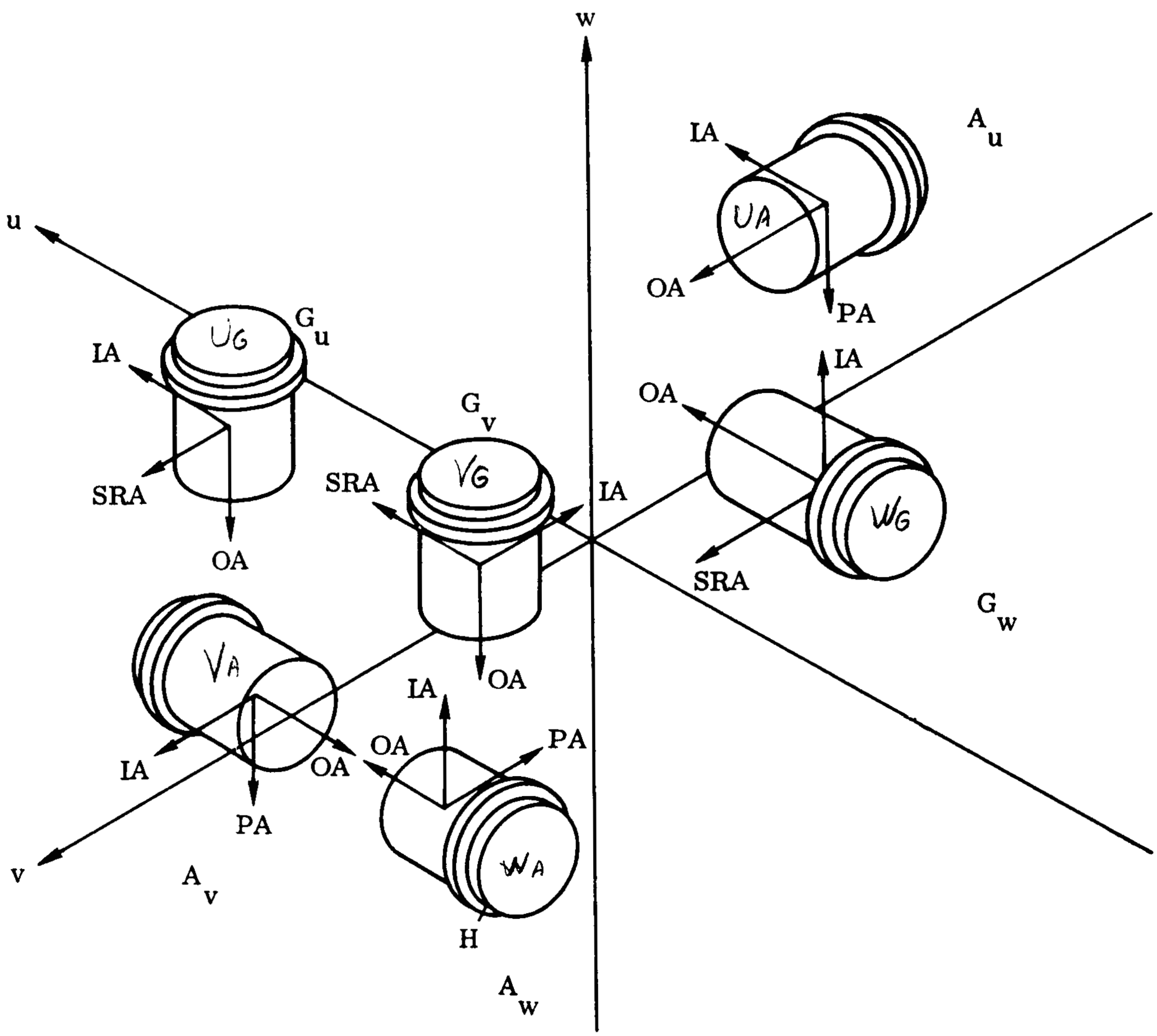
The navigation computer is a serial, binary, digital machine using magnetic drum storage and a high speed integrator for processing the incremental velocity inputs from the pulse-rebalanced accelerometers. It performs the in-flight function of solving guidance equations and issuing steering signals and engine cutoff discretes, as well as generating platform gyro compensation torquing currents for fixed drifts and mass unbalances. It performs the preflight function of aligning the platform, determining the required in-flight compensation coefficients for the inertial instruments in the platform†, and storing the compensation coefficients in its memory. It is composed of two sections, the computer unit and the input-output unit.

The computer drum operates at 6000 rpm and contains 2816 words of permanent storage and 256 words of temporary storage. Each word has a length of 25 binary bits with the leading bit in the case of a data word representing the sign bit and the succeeding bits representing a binary number. The remaining drum space is used for timing, sector address, high-speed integration and arithmetic registers. The inputs from the accelerometers are accepted and integrated in real time (maximum acceptance rate is 3600 pulses/second, which occurs at an acceleration of a little over 11 g's). An 8-word recirculation loop is employed to count down the extrapolated time-to-go until the main engine cutoff discrete is issued.

The guidance program containing instructions and data words is stored on the magnetic drum. It represents the equations which operate upon the input data and calculates outputs for guiding the vehicle. The technique used in developing these equations represent the main body of this report.

1.1 DEVELOPMENT OF GUIDANCE PHILOSOPHY. The starting point for development of guidance equations is the determination of what guidance philosophy to employ

† This task is often termed "Calibrating The Platform."




LEGEND

$A_{u,v,w}$ = ACCELEROMETERS u, v, w

$G_{u,v,w}$ = GYROS u, v, w

IA = INPUT AXIS

OA = OUTPUT AXIS

PA = PENDULOUS AXIS

SRA = SPIN-REFERENCE AXIS

Figure 2. Inertial Component Axis Orientation

for achieving all mission objectives. This effort consists of surveying the mission open-loop trajectory envelopes and trajectory and mission constraints. Equations are then formulated that can most advantageously be used to achieve mission objectives with minimum degradation of payload, and within the limitation of the guidance system input, output, and computational capabilities.

For the Surveyor direct ascent mission, the following guidance criteria and trajectory constraints exist:

1. At injection, the vehicle must have an orbital energy which is a function of the particular launch day and launch time (within the day).
2. At injection, the vehicle's velocity vector must be oriented in pitch and yaw so that the resulting free-fall trajectory intersects the moon. The orientation is a function of the launch day and launch time.
3. At injection, the vehicle's altitude must be closely controlled so that aerodynamic drag, vehicle heating (subsequent to injection), and injected weight are kept within specified bounds.
4. The Atlas/Centaur vehicle must be constrained by the guidance software to fly very close to the prescribed nominal trajectory during the booster and sustainer phases in order to maintain vehicle heating and aerodynamic loading within specified bounds.
5. When the Atlas/Centaur performance is nominal, the effects of guidance software feedback on injected weight loss should be negligible.
6. At injection, the errors (in position and velocity) resulting from the guidance software which satisfy the first five constraints must be an order of magnitude less than the hardware errors.
7. Under the influence of $3\sigma$ Atlas/Centaur performance dispersions, payload loss resulting from guidance software feedback should be negligible.
8. The Atlas booster phase must be terminated by guidance at a specified axial acceleration.
9. The guidance equations are constrained to using, as independent input variables, time and velocity outputs from the accelerometers ("thrust velocity").
10. Guidance software output is limited to a steering vector, selected discretes, telemetry, and platform gyro torquing compensation.
11. The equations must be simple enough to satisfy the vehicleborne computer storage requirements.

To satisfy the guidance criteria and stay within the specified trajectory constraints, several techniques may be feasible. The development of the current guidance philosophy for the direct ascent mission is based to a large extent upon an earlier philosophy developed to satisfy the two-burn parking orbit type of Surveyor mission.

The most important feature conceptually is that of computing a required velocity for the Centaur stages of powered flight.

The required velocity at any point of the trajectory is that velocity which would be required to achieve mission objectives if the powered flight were terminated at that particular point. The required velocity can be determined from explicit, semi-explicit, or polynomial approximations of explicit equations. In general, the explicit equations defining the required velocity tend to be complex and may involve transcendental functions. Because of computer storage limitations and in the interest of minimizing computer cycle time, polynominal approximations in combination with certain simple explicit equations are employed for determination of the required velocity.

The vector difference between the required velocity and the actual vehicle velocity represents the velocity-to-be-gained in order to achieve proper injection velocity. The velocity-to-be-gained is the primary "error vector" from which the steering vector is derived. If the steering vector is computed solely from the velocity error vector, however, it is found that the resultant trajectory deviates in an unsatisfactory fashion from the nominal. Thus, to compute the actual steering vector, an appropriate "trajectory shaping function" is combined with the error vector. The result is a "smooth " well-behaved trajectory which closely agrees with the prescribed nominal; i.e., large, payload-wasting steering maneuvers are precluded.

Another feature of importance in the direct ascent mission is the capability to guide the vehicle through the terminal phase of the booster section of the trajectory and (after the vehicle has passed throughthe region of maximum aerodynamic pressure) through the complete sustainer phase of flight. In these phases, the primary objectives of steering are to "fly" the vehicle satisfactorily through the regions of critical aerodynamic loading and heating, and to initiate, early in flight, corrective maneuvers for Atlas stage performance dispersions. Since a guided Centaur stage is to follow, steering to satisfy explicit guidance criteria during the booster and sustainer phases is not applicable. In this context, primary cutoff of the booster and sustainer phases is based upon performance and structural considerations rather than on velocity criteria.† A simple approach for computing the steering vector during the booster and sustainer phases of flight is to specify the trajectory-shaping function so that the resultant steering vector corresponds closely to the nominal thrust attitude-versus-time profile. The corrective maneuvers are implemented by using a specified function to determine whether or not the vehicle is on the nominal trajectory. If the function senses that the vehicle is "off nominal," appropriate thrust attitude corrections are computed and added to the steering vector.

1.2 <u>DEVELOPMENT OF DETAILED GUIDANCE EQUATIONS</u>. To establish the ultimate set of guidance equations to be used on a particular mission, it is necessary

† Booster cutoff is on axial acceleration; sustainer cutoff is on fuel depletion.

to set up several complex simulations. The first of these is the GD/A simulation (COMBO) of the Atlas/Centaur vehicle and the forces applied to it while moving through inertial space. It represents one of the most versatile, detailed, and accurate three-dimensional powered flight simulations in existence for the Atlas/Centaur vehicles. The Atlas portion has been validated in repeated post-flight analyses during the past five years.

For preliminary guidance equation development, a mathematical model of the selected guidance philosophy is formulated. This model is then programmed for the 7094 computer to be integrated with COMBO so that it can generate closed-loop steering commands for the powered flight simulation. By process of iteration, both manual and automatic, the equations are then "debugged" using simulation runs on the 7094. This technique involves the simulation of both nominal and non-nominal vehicles. The end result is a nearly complete set of guidance equations fulfilling the requirements for steering the vehicle and cutting off engines at the proper trajectory conditions. Since this simulation embodies an ideal computer rather than the actual vehicleborne computer, most of the guidance constrants are preliminary and are currently finalized using an actual vehicleborne computer or a vehicleborne computer simulation.

For finalization of guidance constants and verification of guidance equations, the equations are then written in a codable form for the vehicleborne computer. This effort involves flow charting the equations; scaling the parameters to achieve proper accuracy while precluding the possibility of overflow on $3\sigma$ dispersed flights; providing input for the telemetry format data and its sequencing; and incorporating equations to compensate for guidance platform errors arising from gyro drifts, accelerometer biases, misalignments, and scale factor uncertainties. The product of this task is all of the input information necessary for detailed programming of the equations for the airborne computer.

Coding the guidance program or vehicleborne computer consists of assigning instructions and data words, in chronological order flow-wise, to storage locations cell-by-cell to achieve minimum storage usage while minimizing the number of drum revolutions required to execute the program commands.[†]

Of particular detriment to the final program are equation changes during the course of the coding or after completion of the program. The usual result is inefficient programming demonstrated by increased computer cycle time and in some cases, increased storage requirements.

---

[†] "Computer cycle time" is defined as the number of drum revolutions required to solve the equations once; the quantity is desired to be small so as to avoid autopilot-guidance stability problems.

Having coded the vehicleborne computer program, it is assembled on the 7094 for use in the vehicleborne computer interpretive simulation program. This program is a bit-by-bit simulation of all portions of the Librascope computer excepting the sigmator † portion. The simulation is designed to interpret the guidance program instructions and execute them using the vehicleborne computer mathematical algorithms. Truncation and round-off effects in addition to precision time sequencing corresponding to drum revolution speed are important inclusions.

Using this simulation program, hereafter referred to as LICOS, the guidance equation program is debugged mode-by-mode from flight mode start to flight complete after final orbit injection. This is accomplished by driving LICOS open loop with an acceleration profile derived from a nominal trajectory. Available as output to aid in the validity check of the program is complete telemetry-type information in both binary and decimal form which is compared directly with COMBO reference trajectory data.

After the program is debugged, it is assembled into a combined COMBO/LICOS program called COFLIC. This program provides a simulation of the vehicle and vehicleborne computer operating in a closed-loop configuration. By means of repeatedly running this simulation from launch to injection, a complete checkout of the vehicleborne computer guidance equations program is provided. Final "tuning" of guidance equation constants is also accomplished at this time using a realistic simulation of the actual vehicle working in conjunction with the detailed simulation of the actual vehicleborne computer.

The use of these simulations enables complete checkout of the flight guidance program prior to operating it on the actual vehicleborne computer.

1.3 AN EXAMPLE OF GUIDANCE EQUATION DEVELOPMENT. As an example, consider the task of developing a set of guidance equations applicable to achieving the Surveyor (lunar) mission objectives using the direct ascent mode of operation. The first task of the guidance equations designer is to become familiar with the basic geometry of the open loop powered flight and "free-fall" trajectory. In the case of "earth-fixed" missions (e.g., a ballistic missile flight from an earth-fixed launch site to an earth-fixed target), it is found that one nominal powered flight trajectory defines the mission. Hence, the guidance equations can be designed specifically to satisfy the one trajectory. For the Surveyor mission, however, the target is the moon, and it is found that the trajectory geometry varies with the launch time. Thus, no one nominal powered flight trajectory completely defines the mission.

† The sigmator is composed of special purpose logic and tracks on the computer memory drum; one function of the sigmator is to process the accelerometer data to obtain thrust position and velocity.

The salient characteristics of the trajectory geometry needed for guidance equation design can be developed by assuming that the transfer orbit is a Keplerian ellipse. A typical ellipse, corresponding to a 66-hour (transit time) mission, is shown in Figure 3. The trajectory is elliptical, rather than hyperbolic, since it is not necessary to escape completely free of the earth's gravity field, as would be the case on a Venus mission. A transit time of 66 hours is used because of a mission constraint which requires viewing the lunar impact portion of the trajectory from a specified tracking location.

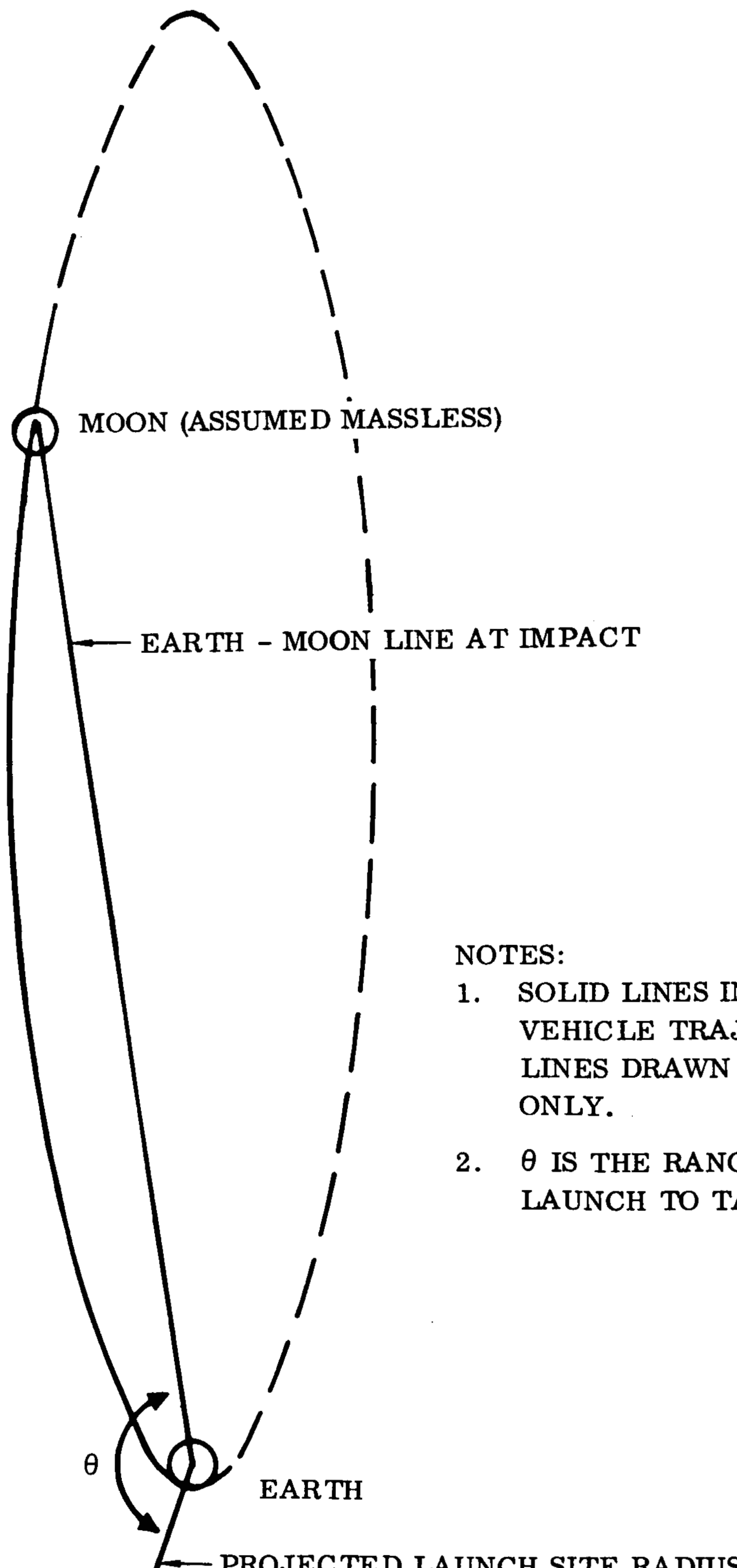
The range angle $\theta$, Figure 3, varies as a function of the daily launch time, as seen in Figure 4. This diagram shows typical trajectories, each launching at a different time during the day, projected onto the unit sphere. The motion of the launch site, caused by earth rotation during the time interval $t_1$-$t_3$, is indicated by the uppermost arc in Figure 4. Note that the motion of the moon during the same time interval describes a much shorter arc. Thus, because of the relative motion between the launch site and target, it is seen that the magnitude of the connecting arcs, $\theta_1$-$\theta_3$, must vary as a function of the launch time, $t_1$-$t_3$.

The range angle $\theta$ is the sum of the powered flight burn arc, $\theta_{pf}$; and the free-fall angle $\theta_{ff}$. Since the powered flight arc $\theta_{pf}$ is relatively inflexible because of payload considerations, the required variation in $\theta$ is achieved by varying $\theta_{ff}$ as a function of launch time.

The technique used to achieve the required $\theta_{ff}$ variation is to shape the powered flight trajectories as a function of launch time such that injection occurs at different locations on a lunar transfer ellipse as shown in Figure 5. For clarity, Figure 5 shows the three trajectories corresponding to launch times $t_1$-$t_3$ oriented in the plane of the paper. The actual trajectories, of course, lie in a different plane corresponding to each launch time. To see this, note that in order to intersect the moon without a payload-consuming plane change, each trajectory must contain the corresponding arc shown in Figure 4.

Although the planar orientation of the lunar transfer ellipse is time varying, the in-plane geometry of the ellipse is relatively stable. To see this, note that the ellipse must intersect the moon (which remains at a relatively constant distance from the earth), and it must also pass through a perigee radius which is constrained to lie within a narrow band of altitudes (90 to 100 nautical miles) because of vehicle performance and heating considerations. The third variable that strongly influences the geometry is orbital energy, which is constrained by time-of-flight considerations. Thus, all lunar transfer ellipses of interest to this discussion have the general appearance of Figure 3.

The important point to observe in Figure 5 is that the injection altitude and velocity vector corresponds to some point on the transfer ellipse. To achieve these variable




NOTES:

1. SOLID LINES INDICATE ACTUAL VEHICLE TRAJECTORY; DOTTED LINES DRAWN FOR INFORMATION ONLY.
2. $\theta$ IS THE RANGE ANGLE FROM LAUNCH TO TARGET.

Figure 3. Typical 66-Hour Lunar Transfer Trajectory

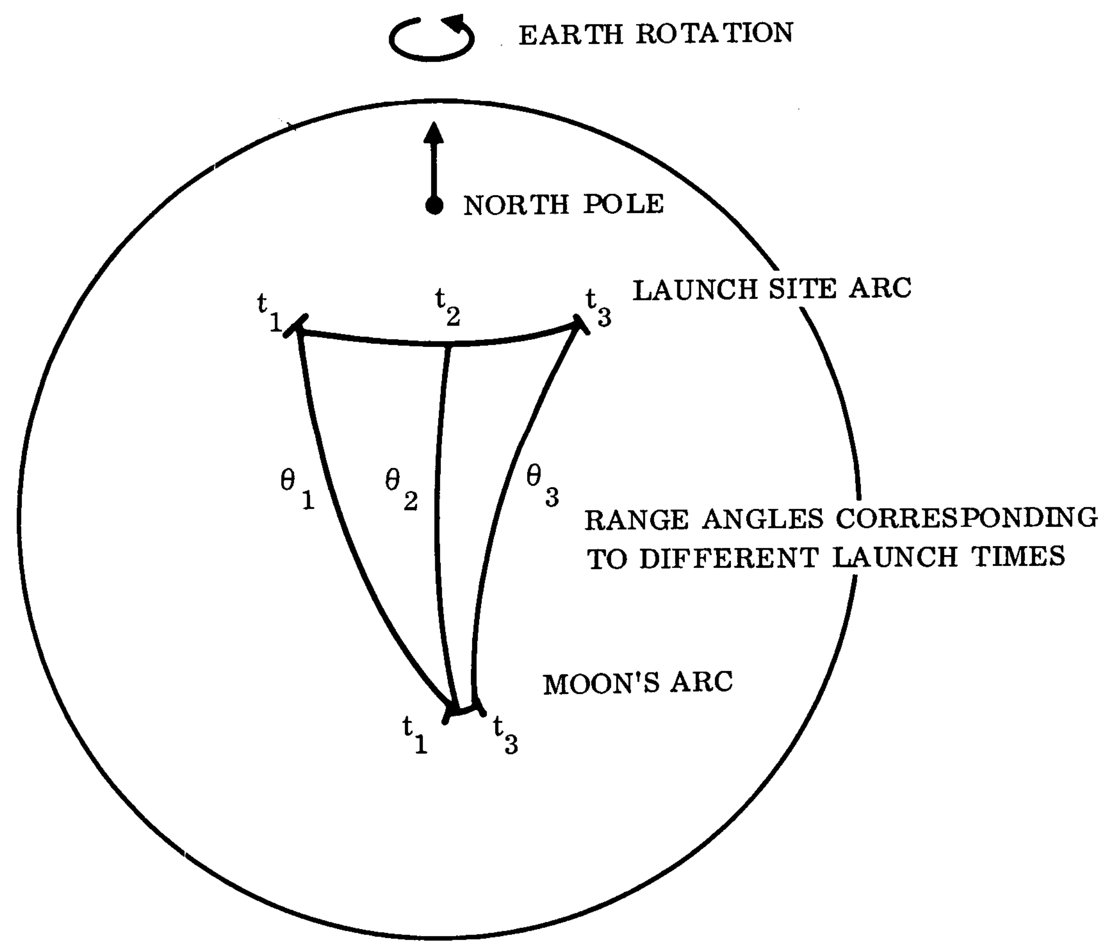



Figure 4. Range Angle Variation Resulting From Finite Launch Time Interval

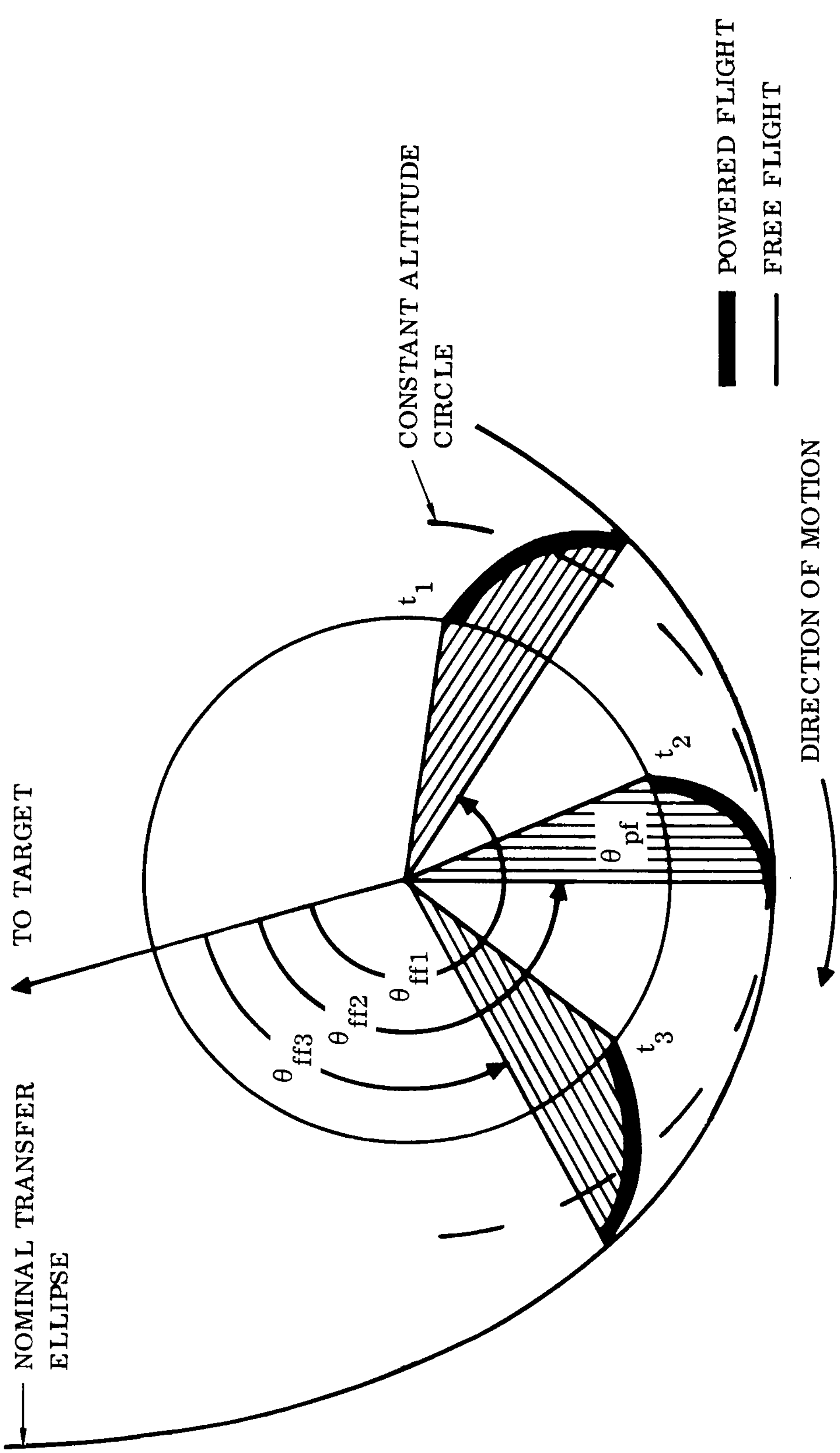



Figure 5. Free Fall Angle $\theta_{ff}$ Variation Using Direct Ascent Powered Flight Trajectories

injection conditions, it is necessary to vary the individual trajectory pitch profiles (i.e., pitch rate vs. time history) as a function of launch time when shaping the open-loop trajectories.

Having reviewed the basic powered flight trajectory geometry, the next task is to design the fundamental guidance equations. If the vehicle behaves as nominally predicted, guidance would not be necessary, and a simple preprogrammed thrust attitude vs. time profile would suffice for a given launch time. However, a properly designed set of guidance equations is necessary for the capability to achieve proper injection conditions on non-nominal trajectories and to incorporate the capability for handling variable launch time.

Three modes of control are available for guidance. These are: the capability to terminate the thrust at any time; and the dual capability to point the thrust vector in both pitch and yaw. To terminate the thrust, an energy criterion is used. Equations are used which compute the actual orbital energy corresponding to the vehicle's position and velocity, compare the actual energy with the desired energy, and issue the thrust termination discrete when a parity occurs. These equations are

$$h = \overline{V}_m \cdot \overline{V}_m - \frac{K_1}{r_m} \quad \text{actual vehicle orbital energy,}$$

$$\epsilon = h_d - h \quad \text{energy-to-be-gained before terminating the thrust, and}$$

$$\epsilon \leq 0: \quad \text{issue cutoff discrete}$$

where

$\overline{V}_m$ is the vehicle velocity vector;

$r_m$ is the position magnitude;

$K_1$ is a gravitational constant;

$h_d$ is the desired orbital energy.

As mentioned earlier, the constant $h_d$ is determined primarily by the time of flight constraint.

The steering vector is used for thrust vector pointing in pitch and yaw. To compute the steering vector, the required velocity, vehicle velocity, and the trajectory-shaping functions are required.

For the Surveyor mission, the target vector concept is useful in defining the required velocity. The target vector is analogous to the earth-moon line, Figure 3, at impact. The direction cosines of this line defines the direction of the target vector in inertial space. Since the inertial platform direction axes are "locked" to the earth until just prior to liftoff, it is apparent from Figures 4 and 5 that the platform direction axes move relative to the earth-moon line prior to launch; hence the coordinates of the target vector vary as a function of the launch time.

The required velocity is defined as that velocity necessary for the vehicle to "coast" on an orbit of energy $h_d$ to lunar intercept, where the direction to this intercept point is given by the target vector.† The required velocity is a function of the desired orbital energy, $h_d$; the vehicle's position magnitude, $r_m$; and the range angle $\theta$ to the target vector as shown in Figure 6.

The target vector is used in the required velocity definition in a dual fashion:

a. The Vector $\overline{V}_r$ lies in the plane defined by $\overline{1}_a$ and $\overline{1}_r$ as shown in Figure 6.

b. The pitch attitude (orientation within the plane) is a function of the range angle ($\theta$) between the target vector and the vehicle's position vector.

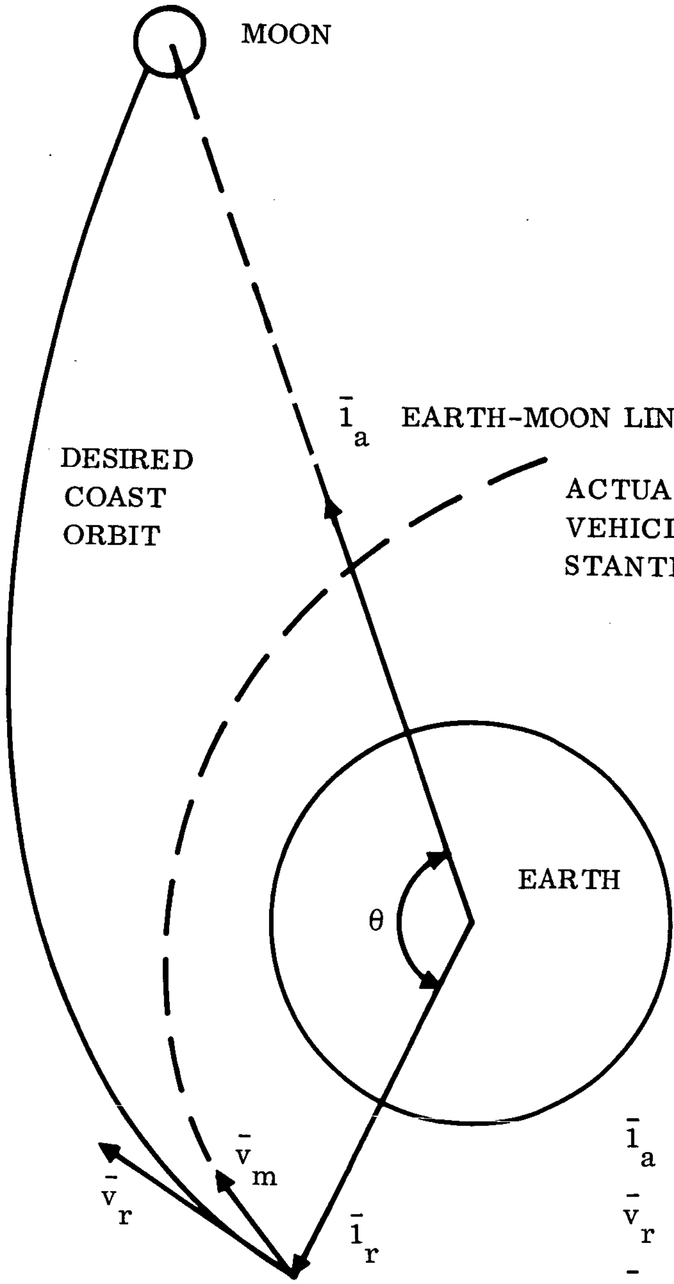
The explicit equation which defines the pitch (or radial) attitude of the $V_r$ vector is highly transcendental. Since the vehicleborne computer does not have specific subroutines for computing arc sines, arc cosines, etc, the equation for the radial component of the required velocity is mechanized in polynomial form:

$$V_{rr} = C_1 + C_2 (r_m - K_2) + C_3 (\sin \theta - K_3) + C_4 (\sin \theta - K_3)^2$$

Here $K_2$ and $K_3$ are the nominal values of the vehicle position and sine of the range angle at the termination of the powered flight, respectively. $C_1$ - $C_4$ are properly determined coefficients, and $V_{rr}$ is the radial value of the required velocity.

At first glance, the use of $\sin \theta$ (rather than $\theta$) in the $V_{rr}$ equation appears to violate the previous comment about the undesirability of using transcendental functions (because of vehicleborne computer limitations). However, it is found that $\sin \theta$ is much easier to compute than $\theta$, as shown in Figure 7. Here are shown the basic tangential, normal, and radial coordinates used in the guidance equations. Note that $\sin \theta$ is obtained by use of a simple vector dot product, which requires three multiplies, three adds, and eleven storage cells in the vehicleborne computer. On the other hand, to compute $\theta$ would require the solution of a series expansion in the vehicleborne computer, a process that is more time and storage consuming than the simple dot product approach.

† In actual use, the target vector is "offset" slightly to allow for all of the perturbing forces which affect the vehicle's free-fall trajectory.




$\bar{1}_a$ - TARGET VECTOR

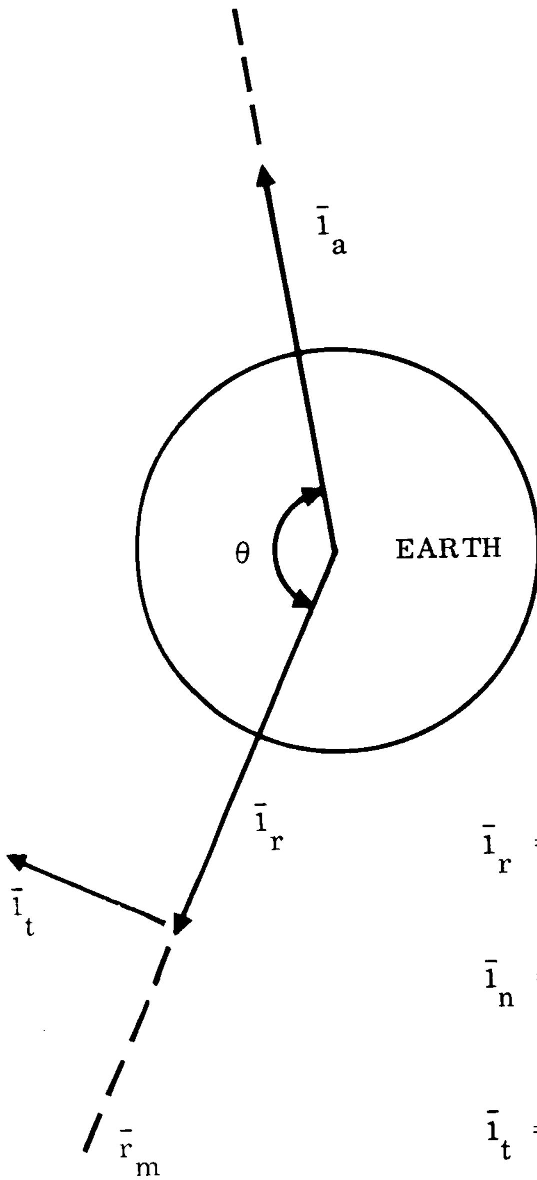
$\bar{v}_r$ - REQUIRED VELOCITY

$\bar{v}_m$ - VEHICLE VELOCITY

$\theta$ - RANGE ANGLE BETWEEN VEHICLE'S POSITION VECTOR AND TARGET VECTOR

$\bar{1}_r$ - UNITIZED VEHICLE POSITION VECTOR

Figure 6. Required Velocity Geometry




$$\bar{i}_r = \frac{\bar{r}_m}{|\bar{r}_m|}$$ UNIT POSITION VECTOR

$$\bar{i}_n = \frac{\bar{i}_r \times \bar{i}_a}{|\bar{i}_r \times \bar{i}_a|}$$ UNIT NORMAL VECTOR

$$\bar{i}_t = \bar{i}_n \times \bar{i}_r$$ UNIT TANGENTIAL VECTOR

$$\mathrm{SIN}\ \theta = \bar{i}_t \cdot \bar{i}_a$$

Figure 7. Guidance Equations Computational Coordinate System ($\bar{i}_t$, $\bar{i}_n$, $\bar{i}_r$)

Note that near the nominal cutoff point, the $V_{rr}$ equation has the desired capability to sense a non-nominal trajectory (i.e., for non-nominal trajectories the parenthetical terms are non-zero), and compute the correct value of the radial required velocity. Thus, it fulfills one desired guidance objective of sensing and correcting for a non-nominal trajectory. The equation is basically a series expansion about the nominal point ($K_2$, $K_3$), however, and as such tends to inaccurate values for $V_{rr}$ as large deviations from nominal are approached. Thus, in order to achieve reasonable accuracy, it is necessary for the guidance equations designer to determine the expected cutoff deviations of non-nominal trajectories in order that a sufficiently large number of terms be included in the equation. On the other hand, an excessive number of terms should be avoided, since they result in storage and computer cycle time penalties.

In the direct ascent mission, the constants $C_1$ - $C_4$ in the $V_{rr}$ expression are dependent on the launch time. To see this, note that the velocity vector associated with each injection point shown in Figure 5 changes as the injection point moves along the ellipse. Being coefficients in a series expansion, the $C_1$ - $C_4$ terms are analogous to partial derivatives which relate changes in $V_{rr}$ to deviations of $r_m$ and $\sin\theta$ from their nominal cutoff values. It is reasonable to suspect that these partial derivative coefficients should be a strong function of the transfer ellipse geometry. This is true in this case. Thus, the variation of $C_1$ - $C_4$ as a function of the nominal trajectory can be decoupled from launch time, somewhat, by defining them with a polynomial expansion which uses $\eta_i$, the injection true anomaly, and $h_d$, the desired orbital energy as independent variables. The true anomaly variable is shown in Figure 8. A typical expansion can be written as

$$C_2 = K_4 + K_5\,\eta_i + K_6 h_d + K_7\,\eta_i^2 + \ldots$$

where the $K_4$ - $K_7$ constants can be determined by least squares fitting procedures. Note that, although explicit dependence of the C constants upon time has been removed, it is still necessary to determine $h_d$ and $\eta_i$ as functions of launch time.

It has been found that the particular equation given for $V_{rr}$ defines it accurately to .5 ft/sec over a range of injection deviations of ±60 nautical miles, downrange. Furthermore, by properly defining the $C_1$ - $C_4$ constants as functions of the trajectory parameters, the equation will yield tolerable accuracy over the range of trajectories including true anomaly variations of ±20 degrees.

To complete the definition of the $\overline{V}_r$ vector, it is necessary to define its normal and tangential component. Figure 7 shows that, by definition, the normal component of $\overline{V}_r$ is 0, since $\overline{1}_n$ is perpendicular to the plane of $\overline{1}_r$ and $\overline{1}_a$. The tangential component can be determined from the magnitude of $\overline{V}_r$. Since the transfer ellipse should have the orbital energy, $h_d$, the magnitude of $\overline{V}_r$ is determined as a function of position

$$V_r^2 - \frac{K_1}{r_m} = h_d$$

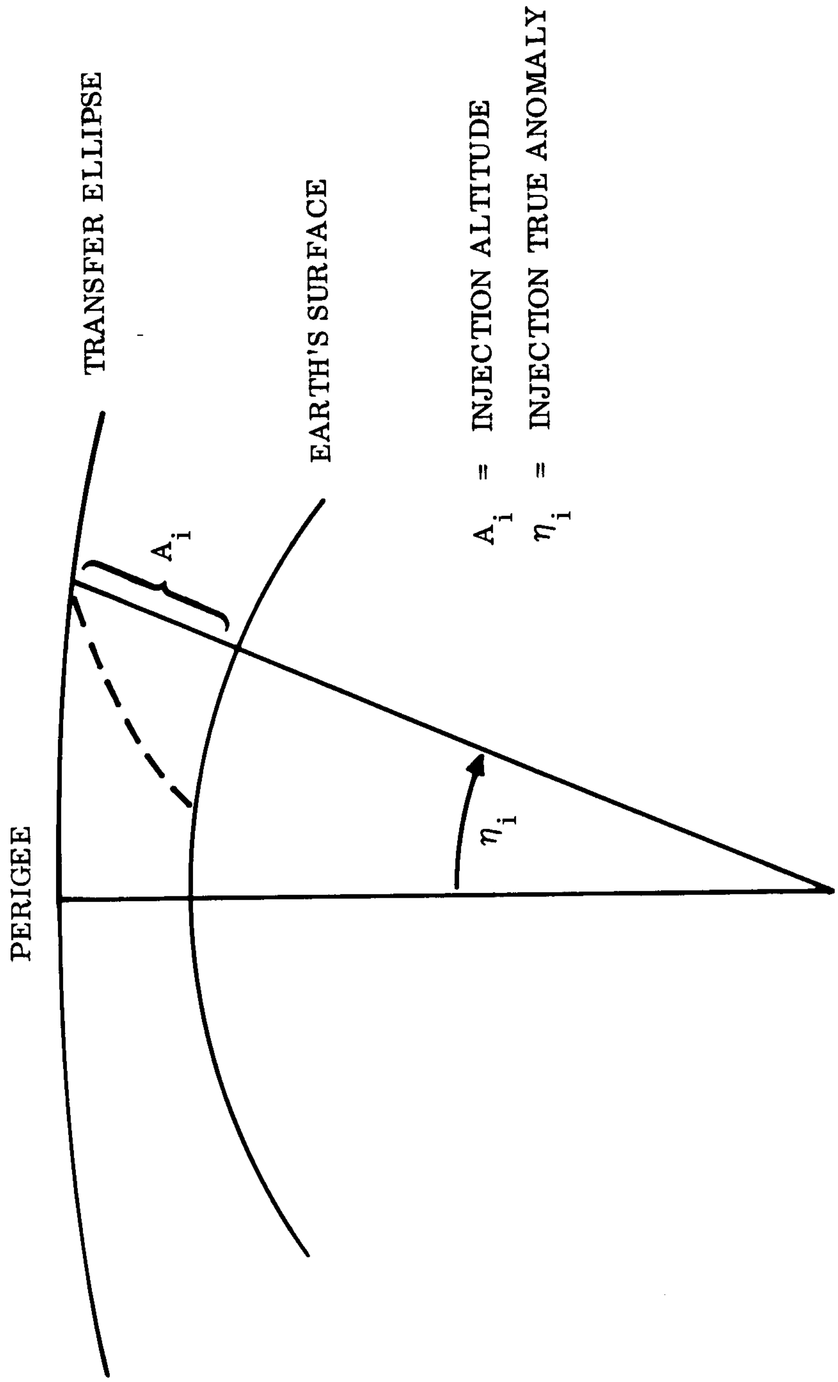



Figure 8. Pertinent Guidance Trajectory Shaping Function Variables

However,

$$V_r^2 = V_{rr}^2 + V_{rt}^2 + V_{rn}^2$$

where

$V_{rn} = 0$ is the normal component of the required velocity, and

$V_{rr}$ is the radial component, stated previously, and

$V_{rt}$ is the tangential component.

Thus

$$V_{rt}^2 = h_d + \frac{K_1}{r_m} - V_{rr}^2$$

To compute the square root, the vehicleborne computer uses the recursive equation

$$V_{rti} = 1/2\left[\frac{V_{rti}^2}{V_{rti-1}} + V_{rti-1}\right]$$

where $V_{rti-1}$ is the previously determined value from the most recent computer cycle. Since $V_{rt}$ is a relatively slowly changing quantity, the recursive relation has been found to yield adequate accuracy.

The vector form of the required velocity is given by the sum of its vector components along the three coordinate directions

$$\overline{V}_r = \overline{1}_t V_{rt} + \overline{1}_r V_{rr} + \overline{1}_n (0)$$

Having $\overline{V}_r$, the velocity-to-be-gained ($\overline{V}_g$) can be computed as

$$\overline{V}_g = \overline{V}_r - \overline{V}_m$$

and the desired thrust pointing direction ($\overline{f}$) is given by

$$\overline{f} = \overline{V}_g + \overline{G}$$

where $\overline{G}$ is the trajectory shaping function.

$\overline{G}$ is used to shape the altitude and radial velocity versus time profile on the nominal closed loop trajectory. These "profile" quantities are influenced strongly by the gravity vector throughout the powered flight. Hence, the most "efficient" direction* for the vector $\bar{G}$ to point is the radial direction. In order for the error vector, $\overline{Vg}$, to be properly nulled at thrust cutoff, the magnitude of $\bar{G}$ should tend to zero as cutoff is approached.

Studies have shown that a simple polynomial function which tends to zero near cutoff can be used to define the shaping function. For the direct ascent guidance equations, the following form is used.

$$\bar{G} = \bar{1}_r C_5 \epsilon^4$$

To develop the appropriate shaping function form, applicable to all nominal trajectories, is one of the prime tasks of the guidance equations designer. This task will be explained in detail, since it illustrates some of the problems peculiar to the software.

Consider the steering vector guidance equation, rewritten in its component form for this discussion

$$\bar{f} = \begin{bmatrix} f_t \\ f_n \\ f_r \end{bmatrix} = \begin{bmatrix} V_{gt} \\ V_{gn} \\ V_{gr} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ G \end{bmatrix} = \bar{V}_g + \bar{G}$$

The problem is to determine the functional form of G and implement it in the vehicle-borne computer with sufficient accuracy. The procedure used to solve for G is based on the principle that the nominal closed loop trajectory should fly very nearly coincident with the nominal open loop trajectory. Now the thrust attitude vs. time history on the nominal open loop trajectory is determined by the thrust vector, $\bar{A}_T$, whereas on the closed loop trajectory it is given by $\bar{f}$. The properly selected G function yields identical trajectories; thus

$$A_{Tt} = f_t = V_{gt} = V_{rt} - V_{mt}$$

$$A_{Tr} = f_r = V_{gr} + G = V_{rr} - V_{mr} + G$$

must hold if the "in plane" trajectories are to be identical. Here $A_{Tt}$ is the tangential component of the nominal open loop thrust vector, and $A_{Tr}$ is the radial component. $V_{rt}$, $V_{rr}$ are the tangential and radial components of the required velocity; and $V_{mt}$, $V_{mr}$ are the tangential and radial components of the vehicle velocity. These two equations may be solved for the required value of G versus time (or versus any other significant trajectory parameter, such as $\epsilon$) to maintain the equality.

$$G_{required} = (V_{rt} - V_{mt}) \frac{A_{Tr}}{A_{Tt}} + V_{mr} - V_{rr}$$

Using the above equation, the nominal open loop trajectory is flown with the guidance equations "piggy back" (i.e., all guidance quantities are computed but not used) to obtain the corresponding values of $G_{required}$ versus $\epsilon$. A typical variation occurring on the direct ascent mission is shown in Figure 9.

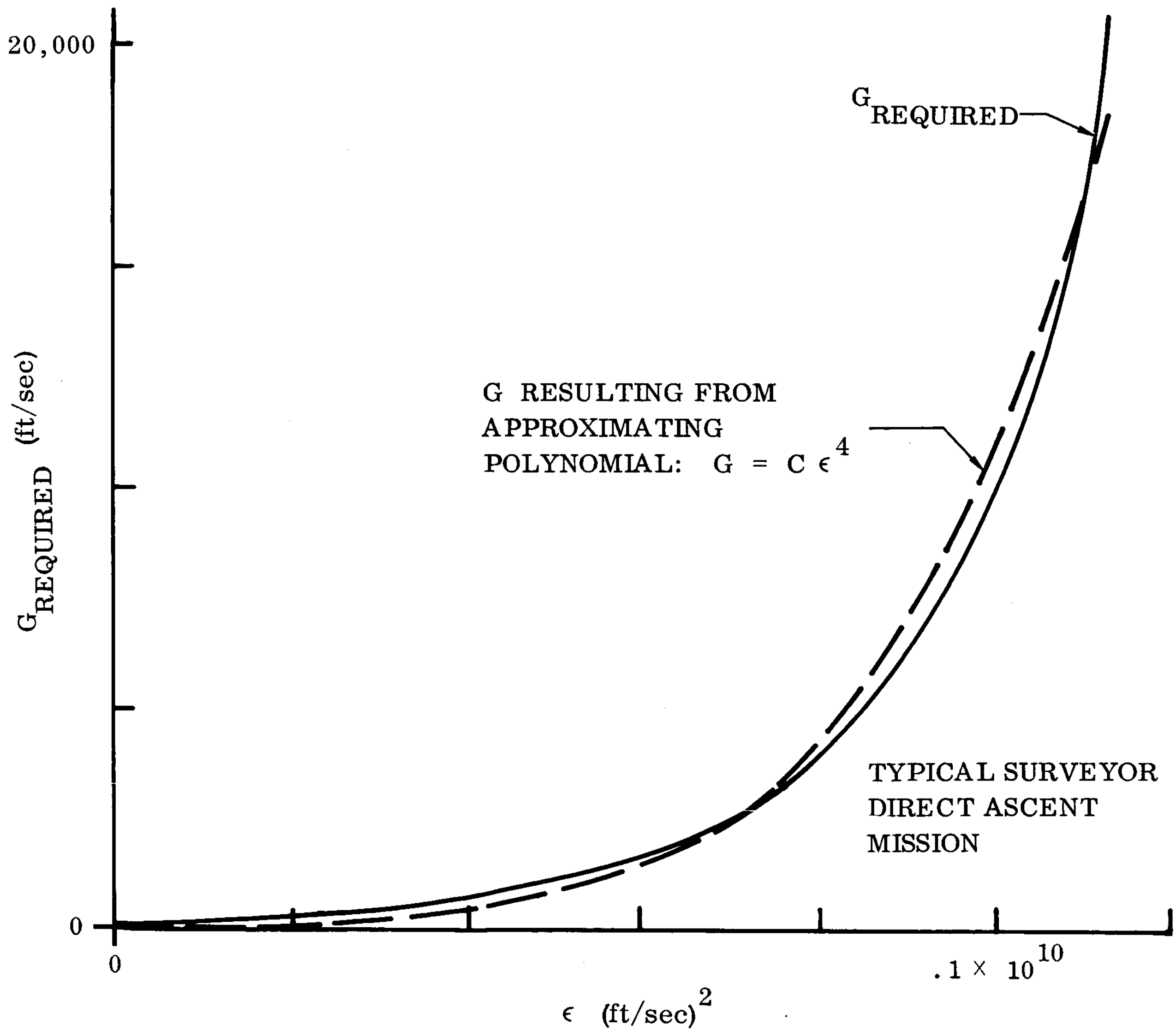



Figure 9. Required Shaping Function Variation for Identical Closed-Loop, Open-Loop Trajectories

The functional form of the actual G which is used in the guidance equations is then determined by the polynomial form which best "fits" the specified $G_{required}$. In choosing the polynomial form there are two opposing considerations: To achieve the best fit, and thus have the closed and open-loop trajectories be very close, a large number of terms would be desirable. On the other hand, storage and computer cycle time limitations point to using a minimum number of terms. Thus, a trade-off is indicated. A typical approximating polynomial curve, obtained from the trade-off study, is also shown in Figure 9.

The foregoing technique must be repeated for each nominal trajectory defined by launch window. In this manner a set of profile functions, each corresponding to a nominal trajectory, is generated. Recalling that these shaping functions are necessary in order that the closed-loop trajectory inject at the specified point on the transfer ellipse, Figure 5, it follows that the functions themselves are dependent on the transfer ellipse geometry. The pertinent variables are true anomaly and altitude, shown in Figure 8. Thus, for each true anomaly and altitude, there corresponds a required shaping function. A set of shaping functions thus generated is shown in Figure 10.

The form of the approximating polynomial must now be suitable for all nominal trajectories (allowing for an appropriate change in the C constant). As mentioned earlier, a quartic in $\epsilon$ satisfies this constraint. The constant $C_5$, however, is a function of the trajectory geometry.

$$C_5 = K_8 + K_g \eta_i + K_{10} A_i + \dots$$

where $\eta_i$ and $A_i$ are the injection true anomaly and altitude, respectively; and $K_8$ - $K_{10}$ are determined by the least squares fitting procedures.

Only a finite number of powered flight trajectories can be used to determine the values of $C_5$. Even if the above "fit" for $C_5$ is accurate at all the fundamental trajectory points, there is no guarantee that satisfactory results will be obtained at intermediate points. To provide this assurance, closed-loop simulations of many additional points must be performed in order to verify the fit.

The correct trajectory shaping function, along with the velocity-to-be-gained function, defines the "core" of the guidance equations. The remaining equations constitute a "support" function, in that they provide values of the basic variables necessary to the "core" equations. These equations are derived by further detailed considerations of the mission constraints, mentioned previously.

Recall that the first constraint stated that the vehicle must have an orbital energy at injection which is a function of the particular launch day and launch time. This constraint implies a launch time dependent $h_d$. Thus, on a given day,

$$h_d = J_1 + J_2 t_L + J_3 t_L^2$$

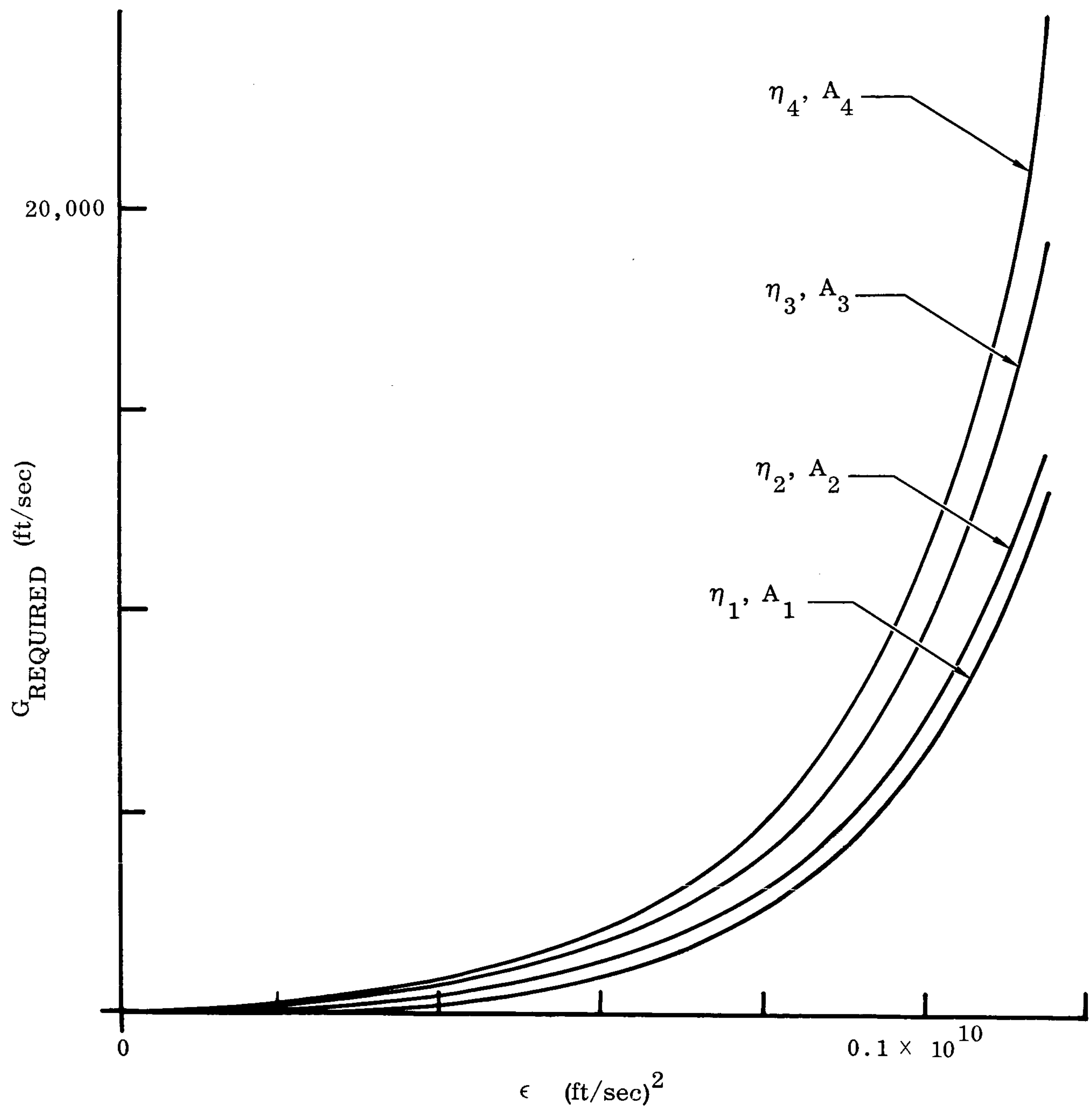



Figure 10. Nominal Trajectory $G_{required}$ Variation With True Anomaly and Altitude

where $t_L$ is the launch time and $J_1$ - $J_3$ are constants of a polynomial fit which defines the required variation of $h_d$ with time. Since the variation of the required orbital energy with launch time is different for each day, the J constants must also be different for each launch day.

The time variable ($t_L$) is particularly important to the "variable launch time" missions. In this context, it should be noted that a liftoff time error of approximately ten seconds can result in significant injection errors. For the Surveyor mission, the vehicleborne computer's time storage cell must be zeroed precisely relative to real time. This can be achieved to an accuracy of .01 seconds with current hardware, and is completed at a known time prior to the start of the launch window on each launch day. Then, when the "enter the flight mode" signal is given to the computer by blockhouse Aerospace Ground Equipment, the program stores the value in the time storage cell into the cell entitled "$t_L$". In this manner, $t_L$ varies with real time throughout the launch window.

The second constraint stated that the vehicle's velocity vector must be oriented in pitch and yaw such that the resulting free-fall trajectory intersects the moon. This constraint is satisfied when $\overline{V}_m = \overline{V}_r$. Recall that direct dependence of $\overline{V}_r$ on launch time was avoided by introducing the variables of the transfer ellipse geometry. However, the basic variables of the injection geometry are launch time dependent, as is the target vector. Thus, on a given day,

$$\eta_i = J_4 + J_5 t_L + J_6 t_L^2$$

$$A_i = J_7 + J_8 t_L + J_9 t_L^2$$

$$\overline{1}_a = \overline{f}\,(t_L),$$

where $\overline{f}$ requires 8 J constants to achieve the desired accuracy.

For reference purposes, the guidance equations which have been formulated in the example are summarized in Table 1. All of the variables which are shown have been discussed previously except for the "basic" position and velocity variables $\overline{r}_m$ and $\overline{V}_m$.

These variables are obtained in the vehicleborne computer by numerically integrating the acceleration due to gravity ($\overline{g}$) and then adding in the thrust position and velocity respectively, to obtain the total vehicle position ($\overline{r}_m$) and velocity ($\overline{V}_m$). Note that, to complete the definition of $\overline{r}_m$ and $\overline{V}_m$, the initial position ($\overline{r}_{mo}$) and velocity ($\overline{V}_{mo}$) of the vehicle at launch must be included.

The equations shown in Table 1 are by no means the complete set of equations necessary to satisfy the trajectory constraints set forth earlier, but are intended to illustrate those equations basic to the software design.

Table 1. Summary of Guidance Equations

| SOLVED ONCE AT LIFTOFF | SOLVED SEQUENTIALLY EVERY COMPUTE CYCLE DURING FLIGHT |
|---|---|
| $\bar{1}_a = \bar{f}_1 (t_L)$ | $\bar{v}_m = \int (\bar{a}_T + \bar{g})\, dt + \bar{v}_{mo}$ |
| $h_d = f_2 (t_L)$ | $\bar{r}_m = \int \bar{v}_m\, dt + \bar{r}_{mo}$ |
| $\eta_i = f_3 (t_L)$ | $h = \bar{v}_m \cdot \bar{v}_m - \frac{K_1}{r_m}$ |
| $A_i = f_4 (t_L)$ | $\epsilon = h_d - h$ |
| $C_1 = f_5 (\eta_i, h_d)$ | $\bar{1}_r = \bar{r}_m / r_m$ |
| $C_2 = f_6 (\eta_i, h_d)$ | $\bar{1}_n = \bar{1}_r \times \bar{1}_a / \lvert \bar{1}_r \times \bar{1}_a \rvert$ |
| $C_3 = f_7 (\eta_i, h_d)$ | $\bar{1}_t = \bar{1}_n \times \bar{1}_r$ |
| $C_4 = f_8 (\eta_i, h_d)$ | $\text{SIN}\ \theta = \bar{1}_t \cdot \bar{1}_a$ |
| $C_5 = f_9 (\eta_i, A_i)$ | $v_{rr} = C_1 + C_2 (r_m - K_2) + C_3 (\text{SIN}\ \theta - K_3) + C_4 (\text{SIN}\ \theta - K_3)^2$ |
| | $v_{rt}^{\ 2} = h_d + \frac{K_1}{r_m} - v_{rr}^{\ 2}$ |
| | $v_{rt} = \frac{1}{2} \left[ \frac{v_{rt}^{\ 2}}{v_{rt}} + v_{rt} \right]$ |
| | $\bar{v}_r = \bar{1}_r v_{rr} + \bar{1}_t v_{rt}$ |
| | $\bar{G} = \bar{1}_r C_5 \epsilon^4$ |
| | $\bar{f} = \bar{v}_r - \bar{v}_m + \bar{G}$ |

NOTE: ALL VARIABLES DEFINED THROUGHOUT TEXT.

Referring to these aforementioned mission constraints, note that the equations do satisfy the first two constraints, which require the vehicle's injection energy and velocity vector to be controlled. However, injection altitude control equations, required for the third constraint, have been developed but are not presented here. It should be noted that this task is another major problem of software design.

The example equations satisfy only part of the fourth constraint in that the G function which was derived maintains the vehicle "close" to the nominal trajectory throughout the sustainer and Centaur phases. Booster phase equations were not discussed, but if derived here, would not necessarily be a simple extension of the sustainer equations because angle of attack control is necessary during the booster phase.

Constraints five through seven require a demonstration of equation performance, from the viewpoint of both payload and accuracy. In order to determine if the equations satisfy these constraints, extensive closed-loop simulations of nominal and non-nominal vehicles are necessary. These simulation runs also can be used to uncover any design defects inherent in the equations, and thus allow design corrections to be instituted before releasing the equations for airborne computer programming.

Since the equations as shown in Table 1, use variables derived from thrust velocity and its integrals, constraint nine is partially satisfied. The other half of constraint nine, concerning computer storage requirements, requires extensive software trade off studies to determine the effects of the number of polynomial terms on equation accuracy.

Note that the example indicated only the logic necessary to issue the Centaur engine cutoff discrete. Other discretes and telemetry were not discussed in the interests of brevity.

# SECTION 2

## GUIDANCE SYSTEM ERROR ANALYSIS†

The results of an error analysis of the Centaur guidance system are usually presented in terms of a figure of merit (FOM), a midcourse correction requirement (MCR), target miss, or as orbit injection errors. The FOM is defined as

$$\mathrm{FOM} \equiv \sqrt{(\sigma_{\dot{x}})^2 + (\sigma_{\dot{y}})^2 + (\sigma_{\dot{z}})^2}$$

where the terms under the radical are the diagonal elements of the covariance matrix

$$\begin{bmatrix} \sigma_{\dot{x}}^2 & \rho_{\dot{x}\dot{y}}\,\sigma_{\dot{x}\dot{y}} & \rho_{\dot{x}\dot{z}}\,\sigma_{\dot{x}\dot{z}} \\ \rho_{\dot{x}\dot{y}}\,\sigma_{\dot{x}\dot{y}} & \sigma_{\dot{y}}^2 & \rho_{\dot{y}\dot{z}}\,\sigma_{\dot{y}\dot{z}} \\ \rho_{\dot{x}\dot{z}}\,\sigma_{\dot{x}\dot{z}} & \rho_{\dot{y}\dot{z}}\,\sigma_{\dot{y}\dot{z}} & \sigma_{\dot{z}}^2 \end{bmatrix}$$

This matrix defines the Gaussian probability distribution of the midcourse velocity corrections required to reduce all target misses to zero. The target misses result from injection errors which, in turn, are computed from the standard deviation (assumed Gaussian) of the guidance hardware error sources. The injection errors are defined as velocity and position errors resulting from particular values of the guidance hardware error sources. Errors resulting from software inaccuracies are included as their distribution becomes defined; in general, these errors are designed to be nearly an order of magnitude less than the hardware errors. For a given set of guidance system error distributions, the FOM varies from trajectory to trajectory. This is the result of the injection errors being functions of the acceleration profile for the duration of powered flight, and the result of the MCR being sensitive to the transfer orbit. (Thus, a given FOM corresponds to a given trajectory and a given set of error source standard deviations.)

The midcourse correction requirement (MCR) is defined as $\mathrm{MCR} \equiv \sqrt{V_x^2 + V_y^2 + V_z^2}$ where the error source values corresponds to the actual measured values as determined from two successive calibration runs of a guidance system. In other words, the MCR indicates the increment of spacecraft velocity required to eliminate the effect of injection errors arising from a particular guidance system, if flown at the time the error source values were determined. The MCR varies from trajectory to trajectory similarly to the FOM. The significant difference between FOM and MCR is that the former represents a number based on statistics of the error sources, whereas the latter is a number based on a particular measured set of error sources.

† See Reference 1 for further details on error analysis technique.

The injection errors are simply errors in position and velocity, in addition to other parameters of interest, which exist as a result of guidance hardware and software error sources. They are normal by-products of the FOM and MCR determinations which occasionally assume importance on satellite-type missions and those missions where midcourse correction capability is not a specified criteria.

To determine the FOM or MCR, the first step consists of determining the injection error covariance matrix resulting from flying the Atlas/Centaur from launch to injection in the presence of guidance hardware errors. To accomplish this, an IBM 7094 program has been written; a general flow diagram of the error analysis program is shown in Figure 11. This program consists essentially of a simulated platform error model, an acceleration profile generator, equations to compute trajectory parameters, coordinate rotation driver, and input and output blocks. The error model consists of mathematical models of the gyro errors, the accelerometer errors, and the influence of initial platform misalignments. Figure 12 shows the inertial measuring unit orientation.

The gyro errors are uncertainties in the fixed torque and g-sensitive drifts. Inasmuch as the platform is torqued in such a way as to compensate for the known drift rates, only the uncertainties in the compensated values lead to platform errors from these sources. The uncompensated fixed torque, or non g-sensitive drifts, cause the platform to rotate at a constant rate about the reference gyro input axis. The direct result is misalignment of platform axes relative to the inertial coordinate system established at launch. This causes the accelerometers to measure accelerations along axes displaced from the desired reference axes. The uncompensated g-sensitive gyro drifts are the result of mass unbalances along the input and spin reference axes of each gyro. A torque about the output axis of a particular gyro will arise from a combination of mass unbalance and acceleration, as shown in Figure 13. As can be seen, a "u" acceleration acting on $m_1$ will cause a torque about the output axis (OA). A "v" acceleration acting on $m_2$ will also cause a torque about the output axis. These torques will cause the platform to drift about the u axis in proportion to the product of mass unbalance and acceleration. The effect when all three gyros are considered is identical to that of the fixed torque errors, that is, the three accelerometers are measuring accelerations along axes other than those desired.

The error model for the gyros is a set of mathematical expressions which when programmed effect platform coordinate system rotation corresponding to platform drift throughout simulated flight. Platform displacements due to the non-g-sensitive drifts are represented by integrals of time multiplied by constant error coefficients, and the displacements due to the g-sensitive drifts by integrals of the product of time and acceleration multiplied by constant error coefficients. Drifts as a result of the elastic property of the gyro are included when of significant magnitude.

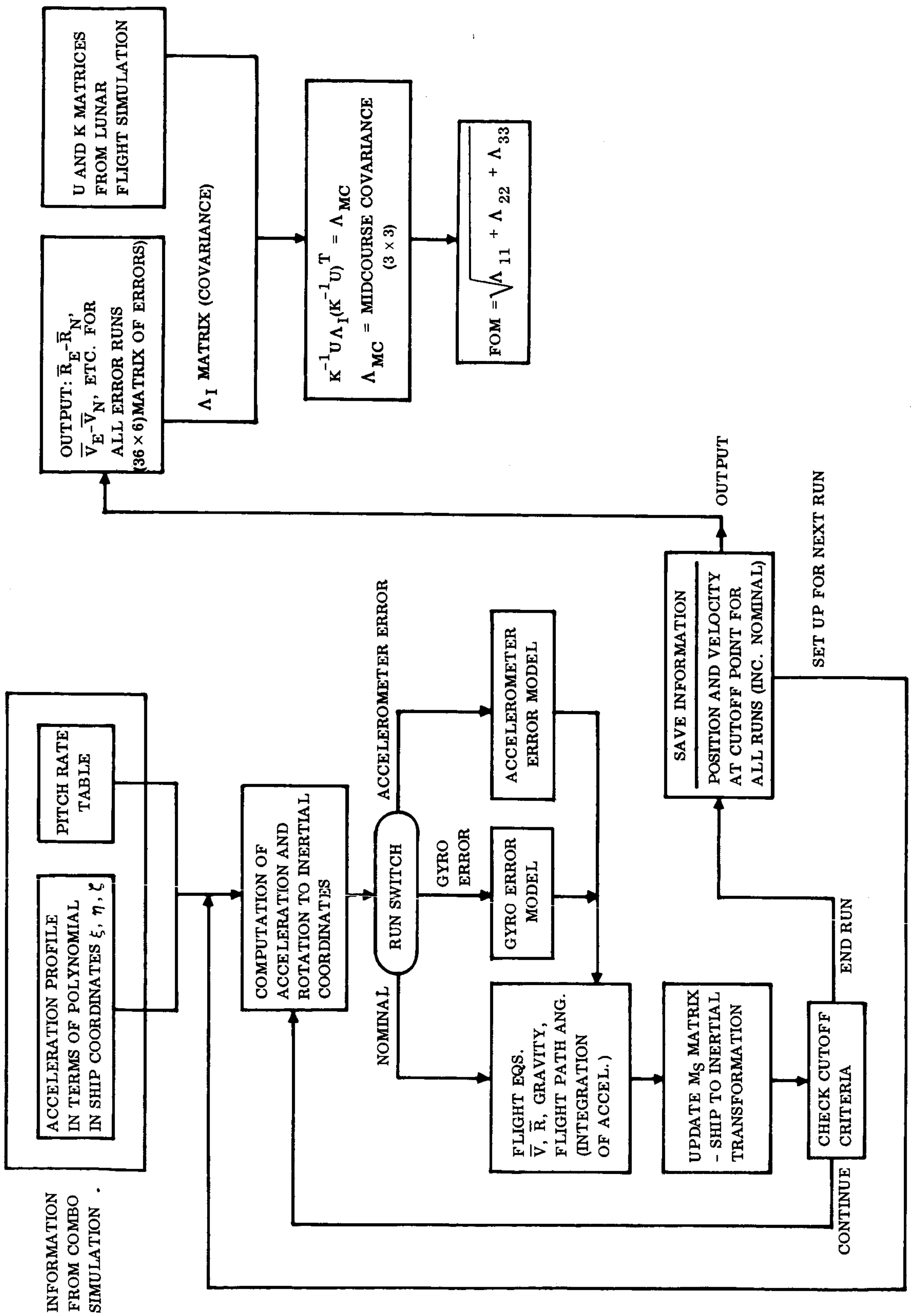
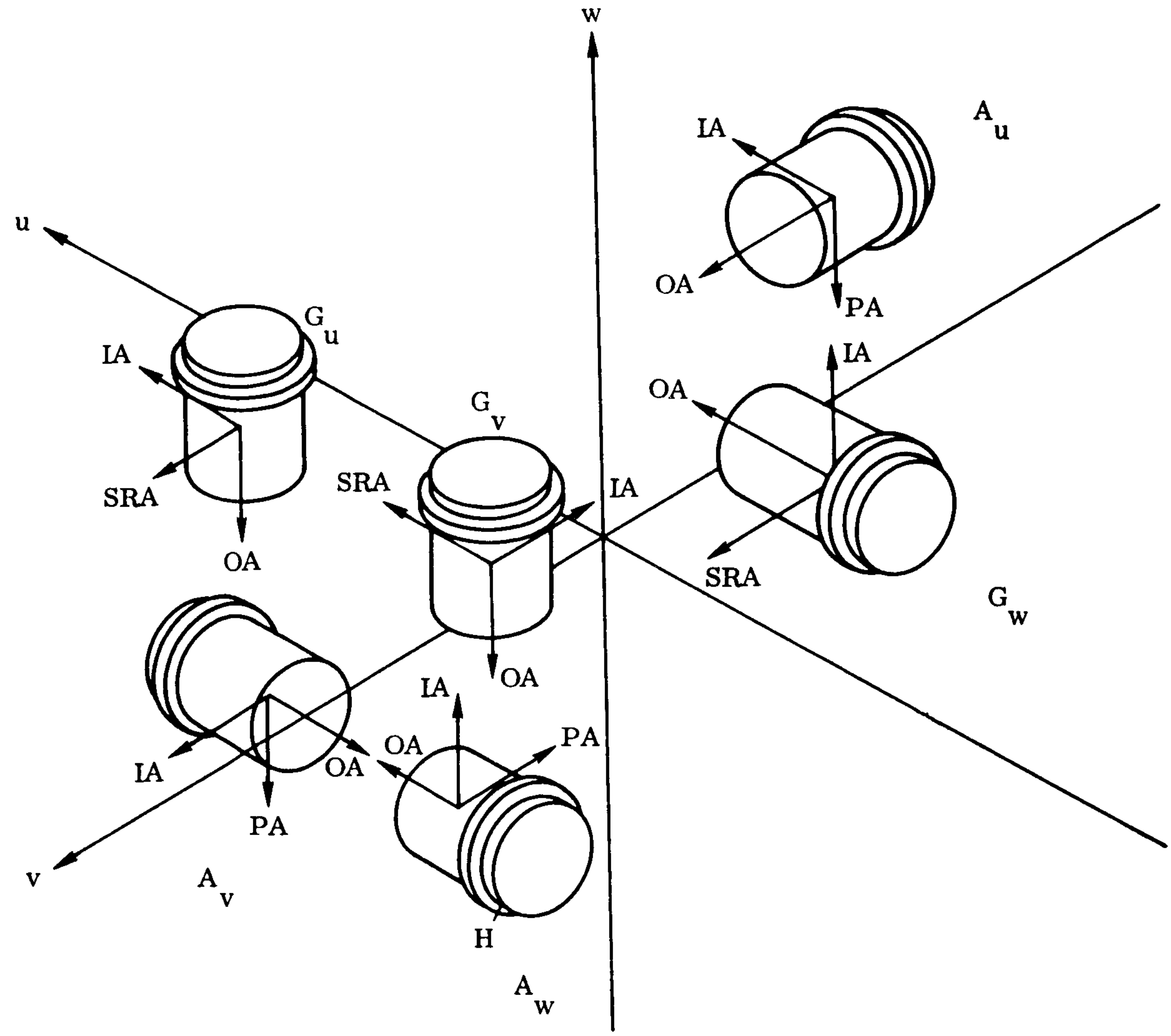



Figure 11. Error Analysis Flow




LEGEND

| | | |
|---|---|---|
| $A_{u,v,w}$ | = | ACCELEROMETERS u, v, w |
| $G_{u,v,w}$ | = | GYROS u, v, w |
| IA | = | INPUT AXIS |
| OA | = | OUTPUT AXIS |
| PA | = | PENDULOUS AXIS |
| SRA | = | SPIN-REFERENCE AXIS |

Figure 12. Inertial Component Axis Orientation

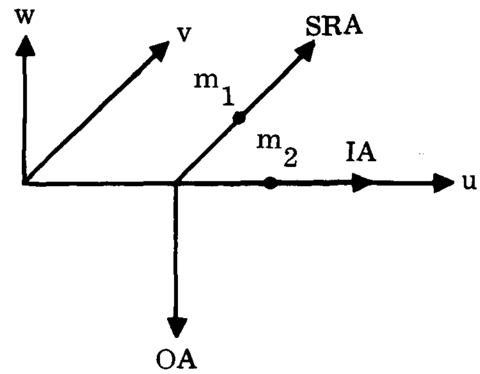



Figure 13. U-gyro Orientation

The accelerometer error model consists of one equation for each of the three accelerometers which contain terms expressing errors in scale factor, bias, and cross-coupling, in addition to higher order terms. The mathematical expression for the total uncompensated error in the u accelerometer is, for example:

$$\Delta A_u = C_0 + C_1 A_u + C_2 A_u^2 + C_3 A_u^3 - C_4 A_u A_w + C_5 A_u A_v$$

where

$\Delta A_u$ = acceleration error along the u axis

$C_0$ = uncompensated bias shift after calibration

$C_1$ = uncompensated scale factor uncertainty after calibration

$C_2$ = second-order nonlinearity coefficient (not compensated)

$C_3$ = third-order nonlinearity coefficient (not compensated)

$C_4$ = cross-coupling arising from misalignment of the input axis about the output axis (not compensated)

$C_5$ = cross-coupling arising from output axis pendulousity (not compensated)

Similar expressions exist for the v and w accelerometers.

Also contributing to acceleration measurement errors are orthogonality misalignments of the input axes of the accelerometers and initial misalignments of the platform relative to the inertial coordinate system. The former are compensated for in the digital computer program except for the uncertainty in the misalignment. The effect of the uncertainty is incorporated in the error model by the addition of linear terms to the accelerometer error model equations.

For example:

$$\delta A_w = C_6 A_u + C_7 A_v$$

where

$C_6$ = uncertainty in misalignment of w accelerometer with respect to u accelerometer

$C_7$ = uncertainty in misalignment of w accelerometer with respect to v accelerometer

The initial platform misalignments appear as a step input in the gyro error model.

The second step in determining the FOM or MCR is to generate K and U matrices. The U matrix is defined as the array of partials of target miss with respect to injection errors for the particular injection-to-target trajectory flown. The U matrix for a particular trajectory is determined by flying a simulated transfer orbit from injection to the target seven times. The seven runs consist of a nominal run and six perturbation runs wherein the six injection errors of interest are perturbed in turn. The results of these runs are used to form the U matrix.

The K matrix is defined as the array of partials of target miss with respect to midcourse velocity corrections. The partials are determined by flying a simulated trajectory from the midcourse correction point to the target four times. One run is a nominal run and the other three represent perturbation runs wherein the components of velocity are perturbed in turn. The results are used to determine the K matrix for the particular trajectory flown.

The K and U matrices, in conjunction with the injection error matrix, are then employed in a computer program for generation of FOM and/or MCR. This program manipulates the matrices in such a manner that the midcourse velocity correction or covariance matrix required to reduce the target errors resulting from the guidance error sources to zero is computed.

Currently, a bimonthly report on guidance accuracy is presented (Reference 1). FOM's and MCR's are tabulated for 17 trajectories which to date have been found to include the worst conditions for propagation of guidance hardware errors. The data presented are based on several types of error source distributions.

One set consists of the current specification standard deviations of the individual errors. From this, FOM's for each of the 17 trajectories are calculated and tabulated.

A second set consists of the present best-estimate of the standard deviations of actual test data error sources. The standard deviations are calculated from all pertinent test data on existing -3 guidance systems. From this data, FOM's for each of the five trajectories are calculated and tabulated. Histograms of MCR's as functions of the actual errors used in computing the error source distribution are also computed and tabulated.

The third set of error source distributions employed consists of specification standard deviations of the -3 guidance system after substitution of GG49H1 gyros for the GG49D15 gyros and GG177B9 accelerometers for the GG116A6 accelerometers. These values are then used in generating FOM's for the five trajectories.

A frequency distribution of MCR's resulting from a Monte Carlo type study will be presented as pertinent. These data will be based on random selection of sets of error values from their corresponding normal distributions, as specified to meet Surveyor requirements. The resulting frequency function can be used to establish effectively the probability of success or failure on the part of guidance system accuracy when the criteria is established as a midcourse correction capability.

# SECTION 3

## POST-FLIGHT GUIDANCE EVALUATION †

Post-flight analysis of the Centaur guidance system is required in order to accomplish the following tasks: (1) Determine adequacy of the guidance equations and constants, (2) Define trouble areas, localized to the smallest group of components or functions, and (3) Establish overall system and component accuracies.

Satisfactory completion of these tasks after each flight points up specific corrective action to be taken for reduction of errors and improvements of the Centaur guidance system. This is the prime objective of the post-flight evaluation effort.

The post-flight analysis in the case of an unsuccessful flight is absolutely essential for isolation of the problem area(s). In the case of a successful flight, it is necessary to uncover undesirable properties which may have led to no overall performance degradation on the particular flight, but on another flight could cause significant performance dispersions.

Analysis of the guidance system will be made from telemetered flight data and from reference trajectory data. The telemetered data will consist of both analog and digital information. The latter is generated by the vehicleborne computer and transmitted in binary bit form, and is, therefore, more accurate than the analog data. A quantitiative analysis will be conducted using, preferably, the digital data. The reference trajectory data, used for comparison purposes, will consist mainly of ground tracking data.

Strictly speaking, the adequacies of the guidance equations and constants are thoroughly verified before flight by many (50 to 100) simulated flights. The information required during flight which will lead to software flight verification is data on vehicle characteristics. Such data will allow a detailed check on the adequacy of the simulation program. If significant differences between the expected vehicle characteristics and the actual properties are found, then the simulation program must be modified. Such modification may, in turn, lead to requirements for guidance equation modification. For actual post-flight analysis of the software, the two vehicle-controlling output quantities of the guidance computer - - (1) the cutoff discretes, and (2) the steering signals - - will be carefully investigated during the post-flight analysis. The actual engine cutoff discretes will be compared with those obtainable from a combined open-loop trajectory/ guidance simulation program. The steering signals will be compared with those derived from a 7094 program driven by the actual flight acceleration profile.

The post-flight analysis of the hardware will consist of three major steps.

† See Reference 2 for further details on the post-flight analysis techniques.

Step A. The analog and digital data will be screened for gross malfunctions, e.g., open circuits or drop-out of excitation power.

Step B. The telemetered output of the Centaur guidance system will be compared with nominal or reference tracking data. The analog data will show, within the limitations of its accuracy, the functional operation of certain areas of the system and the digital data will show the overall performance of the system.

Step C. A quantitative analysis will be made of the Centaur guidance system. The major objective here will be the error separation (or error recovery) program wherein as many guidance system hardware error uncertainties as tracking accuracy permits will be isolated.

Extensive IBM 7094 computer programs have been generated within General Dynamics/Astronautics for accomplishment of Step C and that portion of Step B used in Step C. A basic outline of these programs is presented below.

As stated, one of the objects of post-flight analysis is to provide information regarding trouble areas, localized to the smallest groups of components or functions possible. This holds for the quantitative analysis in particular. For this reason, the system will be broken down to the smallest areas consistent with the data. The operation of such areas will be evaluated on the "blackbox" principle as shown in Figure 14.

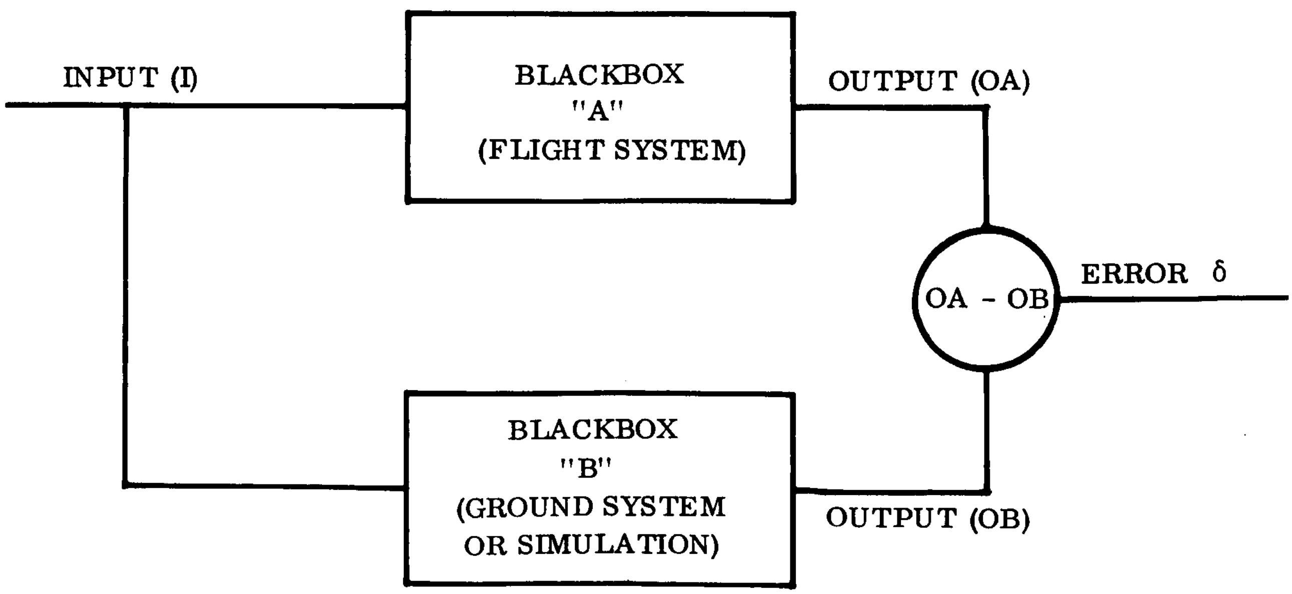



Figure 14. Blackbox Principle

The flight operation of blackbox "A" will be checked by feeding its input, I, into blackbox "B", the ground counterpart of "A" or a simulated "A", and then comparing its output with the telemetered output of "A". The difference, $\delta$ , is a measure of the quantitiative performance of "A".

The larger the number of inputs and outputs available, the greater the number of blackboxes which can be formed (or isolated). Because of instrumentation limitations, the functional breakdown of the guidance system is restricted, therefore, to a relatively small number of blackboxes.

For evaluation purposes, the guidance system will be functionally divided into two major parts:

a. The inertial platform subsystem, consisting of platform gimbal, gyros, accelerometers, and associated electronics.

b. The guidance computer, consisting of input-output unit, memory, and arithmetic section.

As a first step, each of these major parts will be checked on the blackbox principle of Figure 13, i.e., the outputs for known inputs will be compared with the expected outputs as illustrated in Figure 15. The resulting differences will be a measure of the quality of the overall operation of the particular subsystem. The differences, of course, will include errors not attributable to the subsystem being investigated, and where possible these effects will be eliminated during the analysis. The reader is referred to Figure 15 which shows the general flow diagram of the quantitative analysis.

The input to the inertial platform consists of flight trajectory data. The expected output is the data obtained from a reference measuring system such as: (1) a preflight nominal trajectory simulation, and (2) one or more ground tracking systems.

Since other dispersions will result in unwanted a priori differences from the nominal trajectory, the analysis of the inertial platform system will involve mainly the ground tracking systems data. This data will be processed and developed into a "best-estimated trajectory" (BET) program consisting of velocity, position, and acceleration points in addition to the covariance matrices of the tracking system errors.

The thrust velocity derived from the output will be compared with the equivalent velocity measured by the reference system (the velocity resulting from specifc forces is termed thrust velocity). After comparison, the differences measured over the entire trajectory will result in velocity error profiles, which represent an indication of overall performance of the inertial platform. These error profiles will also be employed in an "error separation" or "error recovery" program to isolate specific out-of-tolerance error sources.

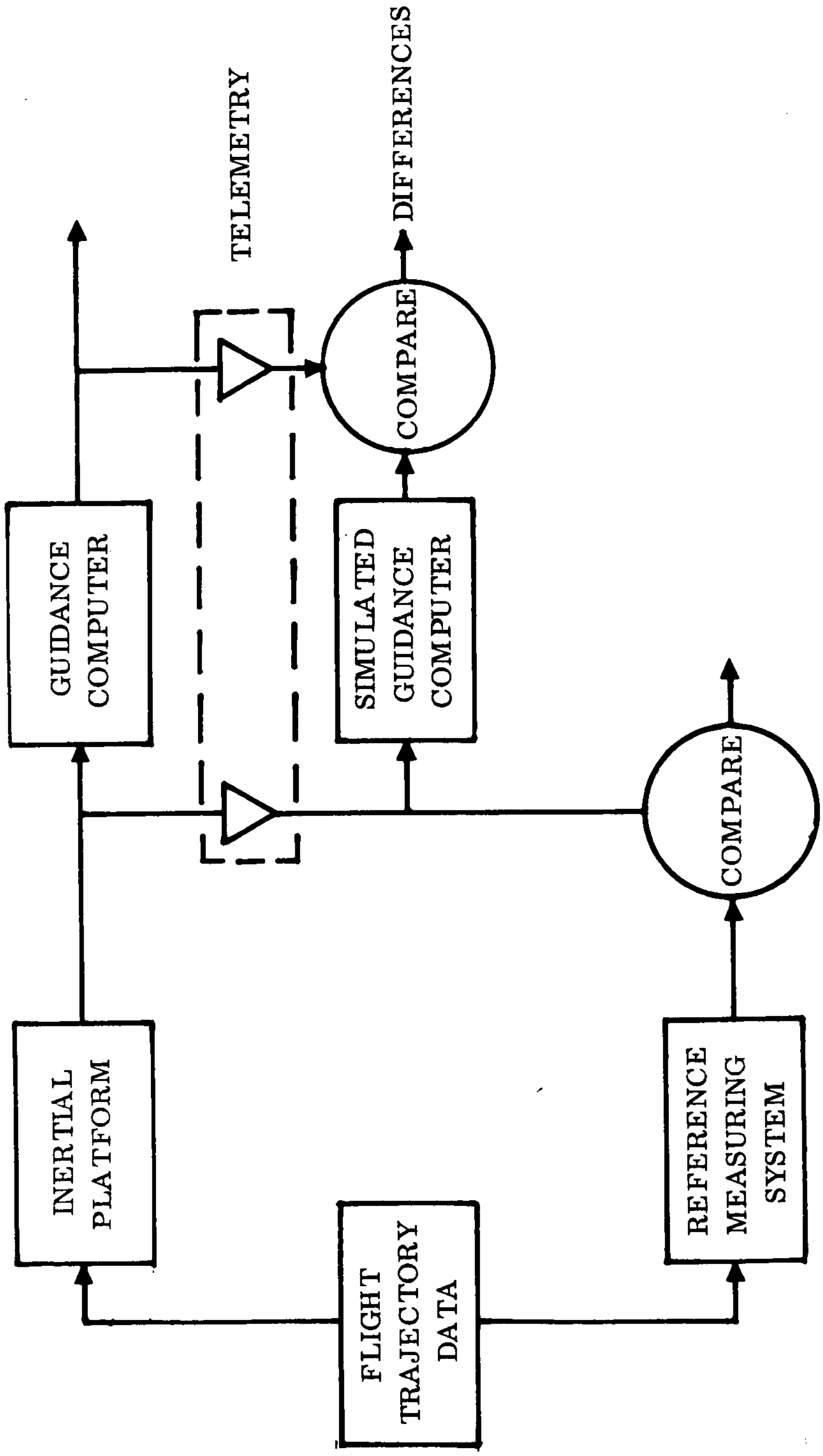



Figure 15. Performance Analysis Breakdown of Centaur Guidance System

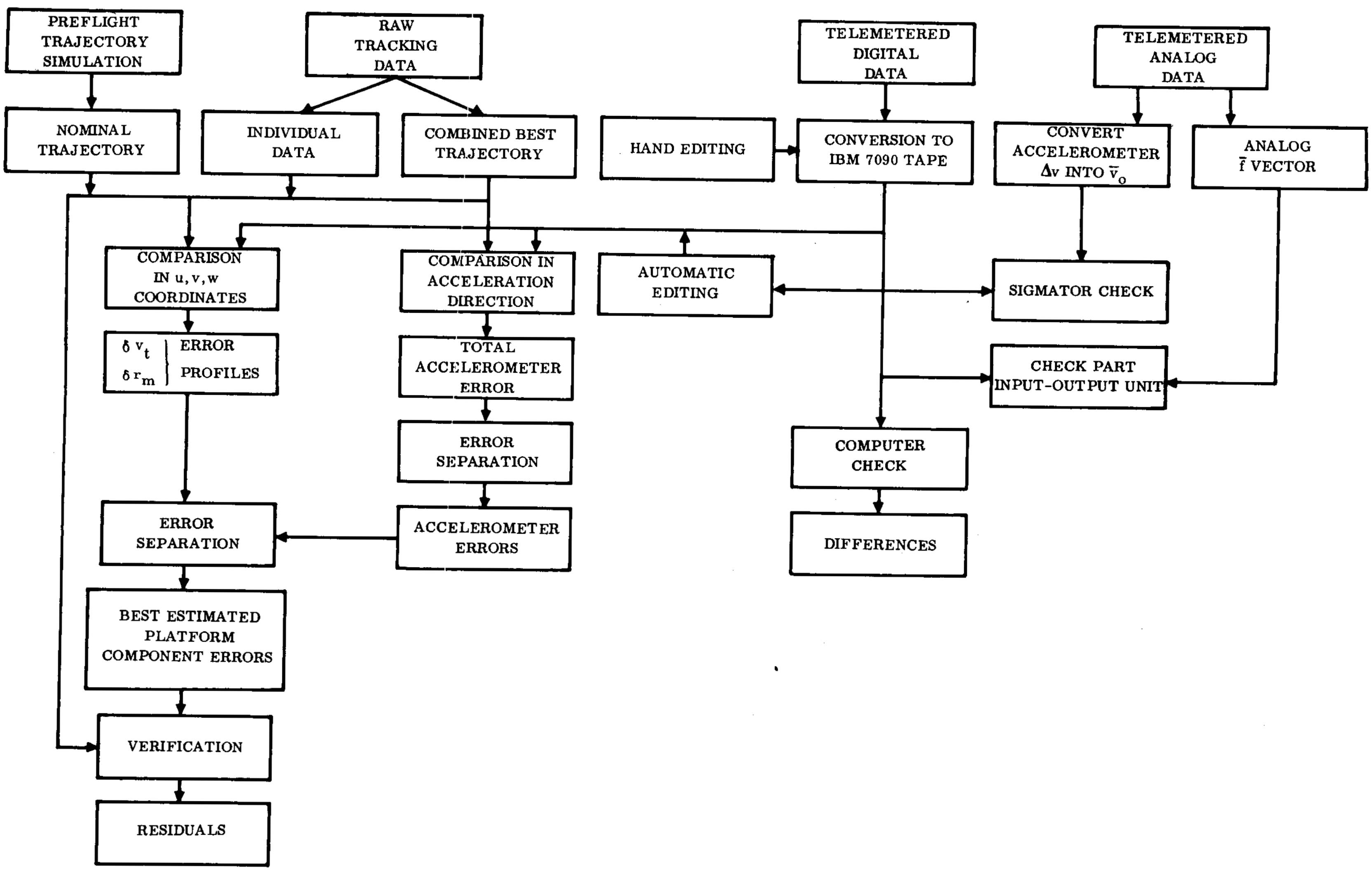



Figure 16. General Flow Diagram of Quantitative Analysis

The output of the inertial platform is the input of the guidance computer. Since this output is telemetered, the data will be used as input to the simulated guidance computer. The output of this simulation will then be compared with equivalent data telemetered from the guidance computer during the flight. The differences will indicate the overall performance of the guidance computer.

The foregoing blackbox check of the two major parts of the guidance system will conclude the objectives noted in Step B.

To satisfy the objective noted in Step C, a detailed analysis, mentioned above, will be made from the velocity error profile curves. The output of the inertial platform system can be represented as a mathematical model of key parameters of the system:

$$\bar{v}_t = \bar{F}\,(p_1, p_2, ----, p_n)$$

where $\bar{v}_t$ is the velocity accumulated from the accelerometers and $p_1$ through $p_n$ represent the system parameters. Differentiation of this expression yields the velocity error

$$\Delta \bar{v}_t = \frac{\partial \bar{F}}{\partial p_1}\,\Delta p_1 + \frac{\partial \bar{F}}{\partial p_2}\,\Delta p_2 + \ldots\ldots + \frac{\partial \bar{F}}{\partial p_n}\,\Delta p_n$$

in terms of the parameter errors $\Delta p_k$.

If we define

$$\bar{\phi}_k = \frac{\partial \bar{F}}{\partial p_k} \qquad \text{and} \qquad \Delta p_k = x_k,$$

then

$$\Delta \bar{v}_t = \sum_{k=1}^{n} \bar{\phi}_k\, x_k$$

Using the subscript T to indicate time of flight along the error profile curve, the velocity error at any time T is:

$$\left(\Delta \bar{v}_t\right)_T = \sum_{k=1}^{n} \left(\bar{\phi}_k\right)_T\, x_k$$

For the mathematical model the derivatives $(\bar{\phi}_k)_T$ are known since they represent the buildup of velocity resulting from a particular type of error. Thus, at any time, T, the velocity error represents a linear expression in $x_k$. If sufficient time points of the error profile curve are considered, a set of linear equations can be found from which the $x_k$'s can be computed.

These $x_k$'s constitute the errors in the parameters of the model and consequently are the error sources of the inertial platform system. It should be noted that such analysis is valid only if the errors remain constant during the flight. Furthermore, because of limitations on tracking accuracy and because some of the errors have equivalent $\bar{\phi}_k$'s, the actual error recovery program, under otherwise ideal conditions, will yield something less than 50 percent of the total number of error sources. Those errors which are isolated, however, represent the significant contributors to guidance inaccuracy.

The functional loops of the guidance computer are not as separable as those of the platform, using the currently available flight data. However, certain areas of the computer system can be analyzed performance-wise.

The operation of the sigmator section of the computer can be checked separately since the input consists of $\Delta v$ accelerometer pulses which will be telemetered. The output of the sigmators is $\bar{v}_\sigma$, which is converted by the computer to $\bar{v}_t$ and telemetered. The telemetered $\bar{v}_t$ will be compared with the output of a ground simulation of the sigmator using $\Delta v$ pulses as input. The differences will indicate the performance of the sigmator section.

Another part of the computer currently capable of being checked is that portion of the input-output unit which handles the steering vector. The steering vector is telemetered from two places: the digital output of the computer and the analog output of the steering potentiometers. By comparing these outputs, the operation of the steering module can be analyzed. Care must be exercised, however, because the loop also contains the signal conditioner and the telemetry system.

This concludes the major post-flight analysis of the guidance system. For further details on the actual evaluation programs, see Reference 2.

# SECTION 4

# REFERENCES

1. Centaur Guidance Systems Bimonthly Report; General Dynamics/Astronautics Report GDA-BTD64-013; 1 February 1964.

2. General Concept of Post-Flight Analysis of the Centaur Guidance System; General Dynamics/Astronautics Report AE62-0288; 15 March 1962.